On Bayesian Analyses of Functional Regression, Correlated Functional Data and Non-homogeneous Computer Models

by

Silvia Montagna

Department of Statistical Science
Duke University

Date: ________________________
Approved:

____
Surya T. Tokdar, Supervisor

____
David B. Dunson

____
Merlise Clyde

____
Joseph E. Lucas

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2013

ABSTRACT

On Bayesian Analyses of Functional Regression, Correlated Functional Data and Non-homogeneous Computer Models

by

Silvia Montagna

Department of Statistical Science
Duke University

Date: ________________________
Approved:

____
Surya T. Tokdar, Supervisor

____
David B. Dunson

____
Merlise Clyde

____
Joseph E. Lucas

An abstract of a dissertation submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2013

Abstract

Current frontiers in complex stochastic modeling of high-dimensional processes include major emphases on so-called functional data: problems in which the data are snapshots of curves and surfaces representing fundamentally important scientific quantities. This thesis explores new Bayesian methodologies for functional data analysis.

The first part of the thesis places emphasis on the role of factor models in functional data analysis. Data reduction becomes mandatory when dealing with such high-dimensional data, more so when data are available on a large number of individuals. In Chapter 2 we present a novel Bayesian framework which employs a latent factor construction to represent each variable by a low dimensional summary. Further, we explore the important issue of modeling and analyzing the relationship of functional data with other covariate and outcome variables simultaneously measured on the same subjects.

The second part of the thesis is concerned with the analysis of circadian data. The focus is on the identification of circadian genes that is, genes whose expression levels appear to be rhythmic through time with a period of approximately 24 hours. While addressing this goal, most of the current literature does not account for the potential dependence across genes. In Chapter 4, we propose a Bayesian approach which employs latent factors to accommodate dependence and verify patterns and relationships between genes, while representing the true gene expression trajectories in the Fourier domain allows for inference on period, phase, and amplitude of the signal.

The third part of the thesis is concerned with the statistical analysis of computer models (simulators). The heavy computational demand of these input-output maps calls for statistical techniques that quickly estimate the surface output at untried inputs given a few preliminary runs of the simulator at a set design points. In this regard, we propose a Bayesian methodology based on a non-stationary Gaussian process. Relying on a model-based assessment of uncertainty, we envision a sequential design technique which helps choosing input points where the simulator should be run to minimize the uncertainty in posterior surface estimation in an optimal way. The proposed non-stationary approach adapts well to output surfaces of unconstrained shape.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

$\mathrm{Ga}(\alpha, \beta)$	Gamma distribution with shape parameter α and rate parameter β.
$N(\mu, \sigma^2)$	Univariate normal distribution with mean μ and variance σ^2.
$N_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	k-dimensional normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
$\mathrm{Unif}(a, b)$	Unifrom distribution on the interval (a, b).

Abbreviations

FPCA	Functional principal component analysis.
LFRM	Latent factor regression model.
MCMC	Markov Chain Monte Carlo.
MGPSP	Multiplicative gamma process shrinkage prior.
PCA	Principal component analysis.
ROC	Receiver operating characteristic.

Acknowledgements

To begin with, I would like to express my deepest thanks towards my advisor Surya Tokdar for his continued advise, help, and encouragement. I have deeply enjoyed our conversations and I am most thankful for the generous time and attention he dedicated to me throughout my time at Duke.

I would like to thank my thesis committee members, Merlise Clyde, David Dunson, and Joe Lucas, for their time and support. They have provided extremely valuable ideas, critiques, and guidance. I would like to further thank David Dunson, my first year adviser, for his constant interest and support throughout my entire time at Duke. I would like to thank Brian Neelon for his encouragement and many useful and enjoyable discussions. It has been a great pleasure working with Irina Irincheeva over the last few months and I would like to thank her for the helpful suggestions and feedback. A special and most heartfelt thanks goes to Igor Prünster for always being there as a mentor and friend.

1

Introduction

The rapid evolution of data collection technologies over last two decades has permitted vast quantities of data to be recorded densely over time or space. These data are usually regarded as error-prone measurements of smooth functions which are assumed to underlie and generate the observations, thus the acquired name of functional data. There is clearly a need for approaches for flexible dimensionality reduction and discovery of sparse latent structure underlying these very high-dimensional data. A wide array of methods has been developed for the analysis of diverse types of functional data (e.g., curves, surfaces, images, etc.), but open problems remain. Within the general framework of Bayesian statistics, this dissertation looks at three main themes, each illustrated by a specific application.

The first theme is that of functional regression. Often, there is interest in studying the relationship between functional predictors and response variables. For example, the functional predictor in an epidemiological study could correspond to the trajectory in the level of gestational blood pressure, while the response corresponds to some pregnancy outcome, such as birth weight. In such cases, there is substantial interest in building flexible joint models for relating the functional predictor to a scalar, vector-valued or functional response, adjusting for covariates. The goal needs to be addressed while providing a low-dimensional representation of the functional predictor and, potentially, the functional response.

The second theme is that of dependent functional data. The rapid progress that the analysis of functional data has experienced in the past decades has lead to more complex structures being amenable to such techniques. In particular, there is need for methods that can deal with the complex correlation structure present across many functional data. Our applied motivation presents such feature of dependence, and concerns the analysis of DNA microarray datasets for which time-course gene expression trajectories are being treated as functional data. The goal is the detection of circadian genes that is, genes whose expression curves exhibit a periodic pattern with a period of approximately 24 hours. At any given time point, it is natural to expect that the expression level of one gene might depend to some extent to the expression of the other genes. Therefore, the correlation structure must be explicitly taken into account.

The third theme is the analysis of computer models. This is a particular type of functional data in that the output of such models is a function over the space of inputs of the computer model. Most often, this output is a function of time or a surface. The focus is on the modeling of non-homogenous computer models, namely functions which exhibit abrupt local features, while accommodating model-based assessment of uncertainty to be used for sequential design.

There is one common idea underlying different topics, namely the use of latent modeling within the general framework of Bayesian statistics. However, the idea practically assumes very different flavors through themes. It ranges from latent factor modeling for low dimensional summary, accommodation of dependence and joint modeling in functional regression to the use of latent Gaussian processes as unobserved inputs for the modeling of non-homogenous computer models. In the remainder of this Section, we provide a thorough description of the three main topics of this dissertation together with relevant literature review and presentation of applied motivations.

1.1 Statistical analysis of functional regression

Many modern statistical analyses involve variables best represented as curves, surfaces or more general functions (Ramsey and Silverman, 2005). Examples include biomarker trajectories, images, videos, genetic codes and hurricane tracks. Data on such curves may come into two flavors, either measured on a dense, regular grid common to all observation units (subjects) or as measurements taken at irregular time points or locations that vary from subject to subject. Analyses of these two kinds of data are labeled, respectively, functional and longitudinal data analyses (Rice, 2004). In Chapter 2, we explore the important issue of modeling and analyzing the relationship of such data with other covariate and outcome variables simultaneously measured on the same subjects.

The applied context that partially motivates some of the methodological and computational work presented in Chapter 2 concerns the so-called Healthy Pregnancy, Healthy Baby (HPHB) study, an ongoing prospective cohort study examining the effects of environmental, social, and host factors on racial disparities in pregnancy outcomes. The specific aim of our study is to characterize the longitudinal trajectory of blood pressure, considered over the entire course of pregnancy, while simultaneously addressing three main objectives: i) obtain a low dimensional representation of the individual curves; ii) incorporate covariate information (e.g., maternal age, maternal race, parity), thus allowing the distribution of the curves to change flexibly with predictors; and iii) link the clinical and functional predictors to subsequent health responses (e.g., gestational age at delivery, birth weight). Because functional data are infinite dimensional, their statistical analysis necessitates obtaining a low dimensional representation of the individual curves. Therefore, objective i) becomes absolutely crucial for building a hierarchical model where the curves are to be related to other covariates recorded on the same subjects.

The existing literature on functional data analysis and longitudinal data analysis does not offer an encompassing framework that can address simultaneously the three aspects mentioned above, though there is a rich array of methods for each individual task. The most widely used tool to represent curves through a low dimensional vector is functional principal component analysis (FPCA) (Rice and Silverman, 1991; James et al., 2000; Yao et al., 2005). In FPCA, a finite number of basis functions are derived by eigendecomposition of a smoothed version of the empirical covariance function of the observed curves. Each curve is then represented by a vector of eigen-scores with respect to the estimated basis. These scores are used to build a two-stage, plug-in model of how the curves affect the response variable. Crainiceanu and Goldsmith (2010) propose a refinement where they plug-in only the functional principal component analysis basis functions at the second stage, while jointly modeling the eigen-scores with other variables of interest.

However, there is very little literature on how to perform FPCA when the curves may depend on additional covariates. Jiang and Wang (2010) recently proposed an extremely flexible approach that accommodates covariates, but their method faces serious practical difficulties when the covariate dimension is not minuscule or when different covariates have a different degree of influence on the curve.

As an alternative to plugging-in FPCA bases and/or scores, which might underrepresent uncertainty, one can directly build models on the space of curves and then use discriminant analysis to perform functional classification. However, existing methods of this kind (De la Cruz-Mesia et al. (2007) and Dunson (2010) from a Bayesian standpoint) do not include covariate information to model the curves, and an extension along this line appears challenging in absence of a sparse representation of the curves. It is also possible to completely ignore modeling of the curves and just build regression models for scalar outputs based on functional and non-functional covariates (Reiss et al., 2010; Zhu et al., 2011). Such approaches face difficulties when predictions are to be made with the functional covariates only partially and sporadically observed, such as when predicting the possibility of a low birth weight delivery given 5 MAP measurements until the 30th week of pregnancy. Additional references on functional regression in a Bayesian context include Behseta et al. (2005); Ray and Mallick (2006); Dunson (2009); Petrone et al. (2009); Rodriguez et al. (2009); Bigelow and Dunson (2009).

In very different approaches, Nagin (1999) and Jones et al. (2001) adopt a mixture model representation to characterize curves through latent classes and let covariates impact on the class probabilities. However, they consider the curves only as response variables and do not discuss models where the curves play the role of functional predictors. Potentially, their method can be extended to an encompassing framework like ours by letting the latent class impact the distribution of the response variable, but this extension was not addressed by the authors. Secondly, by representing each curve by a vector of scores (instead of a single group label), we allow other variables to influence or depend on the curves in a local way. Alternatively, James and Sugar (2003) propose a model for clustering sparsely sampled functions assuming either a classification or mixture likelihood, but no attempt is made to build response models.

We propose a new Bayesian latent factor model for functional data characterizing the curve for each subject as a linear combination of a high-dimensional set of basis functions, and place a sparse latent factor regression model on the basis coefficients. Within our framework, it is possible to study the dependence of the curve shapes on covariates incorporated through the distribution of the latent factors, and we can accommodate the joint modeling of functional predictors with scalar responses or multiple related functions. We avoid two-stage procedures by building a framework that simultaneously accommodates function-on-scalar and vector-on-function regression. Also, our model preserves the modeling goal of FPCA, that is, identifying a common basis and assigning low dimensional scores to individuals with respect to this basis.

1.2 Statistical analysis of correlated functional data

Circadian rhythms are biochemical and physiological functions that display an oscillation of approximately 24 hours. Examples of circadian rhythms in humans include sleep-wake cycles, hormone production, blood pressure and body temperature. These rhythms represent fundamental adaptations of an organism to light/dark cycles due to Earth rotation around its own axe. Circadian rhythmicity of genes and their protein products has been object of active research in both animals and humans (Edwards et al., 2006; Dodd et al., 2007; Phillips, 2009; Jouffe et al., 2013), and a range of clock genes has been identified in almost all species. The regulation of circadian rhythms is a complex molecular mechanism that involves the central and peripheral nervous system, and gene expressions in the brain and all around the body (Phillips, 2009). For example, a range of rhythmically expressed genes can control the cell-division cycle (Matsuo et al., 2003), which is a fundamental process in most organisms. The disruption of circadian rhythms has been linked to a variety of pathologies in humans. In particular, the International Agency for Research on Cancer reports “shift-work that involves circadian disruption is probably carcinogenic to humans” (Straif et al., 2007). Therefore, there is a considerable interest in the identification of genes that control the timing of many physiological processes, and the scientific interest calls for statistical models suitably designed to detect periodic pathways among a very high number of gene expression profiles (curves). Ideally, methods would allow building of a full joint model that allows each gene to have its own trajectory, while accommodating dependence in these trajectories across genes to allow for inherent synchronism or asynchronism of the curves.

Several authors have tackled this periodicity detection problem in biomedical research over the last couple of decades. Chudova et al. (2009) give an excellent review of the main existing techniques, which can broadly be classified as time domain or frequency domain analyses. Time domain methods are essentially pattern-matching techniques: cosine curves of varying periods and phases are fit to each gene expression profile or "transcript" separately and the best fit to the experimental data is retained to describe the signal (Straume, 2004; Hughes et al., 2010). Pattern-matching methods are simple and computationally efficient, but not very effective at finding periodic signals that are not perfectly sinusoidal (Chudova et al., 2009). Frequency domain approaches combine spectral analysis with multiple hypothesis testing (Wichert et al., 2004; Ahdesmäki et al., 2005). Specifically, one obtains the spectrum of an expression profile, and the hypothesis of significance of the dominant frequency is tested against the null hypothesis of absence of periodic signal. The analysis is carried out probe by probe independently and the obtained significance values are corrected for multiple testing using methods such as Bonferroni, Benjamini-Hochberg or others. Chudova et al. (2009) remark that frequency domain methods are most effective on long time series. However, this is not a typical feature of circadian studies, which are usually designed to collect data every 2 or 4 hours over two circadian cycles (48 hours). Therefore, coarse sampling and short periods of data collection are typical features of these studies. Chudova et al. (2009) propose a Bayesian mixture model for the identification of patterns of unconstrained shape. The authors claim that existing computational methods are biased toward discovering genes whose transcripts follow sine-wave patterns. Instead, the focus is on the discovery of circadian regulated genes with non-sinusoidal transcripts.

As a separate line of research for the analysis of genomic data, several model-based clustering algorithms have been proposed in both the classical and Bayesian framework (Yeung et al., 2001; Luan and Li, 2003; Wakefield et al., 2003). However, clustering algorithms need to be customized to reflect the scientific interest of identification of circadian genes. In particular, efforts should concentrate around refining clusters that contain potentially interesting genes while no time should be wasted on finding an optimal partition of obviously non-circadian genes. In this regard, Anderson et al. (2006) use the algorithm in Heard et al. (2006) many times on various partitions of the genes with a Fourier basis to extract rhythmically expressed genes. A score is calculated for each partition of the genes and the score determines the clustering.

A key assumption made in all the approaches above is that of independence of the genes. The clustering algorithm described in Anderson et al. (2006) assumes dependence at a cluster level only whereas clusters vary independently. However, the assumption of independence is often too strong to be realistic in many applications. The data set that motivates our work is a recent microarray experiment designed to assess whether the circadian clock might coordinate translation in mouse liver (Jouffe et al., 2013). Two mice were sacrificed every two hours over 48 hours and 3 μg of polysomal and total ribonucleic acids (RNAs) from each animal were pooled to quantify the expression of each gene. It is natural to expect that the expression measurement for gene i at time j might depend to some degree on the expression of the other genes measured at the same time. Therefore, our motivating application is a typical example of functional data set where the correlation structure across functional data, here the gene expression trajectories, must be explicitly accounted for.

The goal of our analysis is to identify periodic signals in circadian experiments via a flexible Bayesian approach that accounts for the correlation structure in the data. Essentially, we decompose the true, de-noised underlying signal for each transcript as a series expansion of sine and cosine curves to extract rhythmic signals, while we accommodate for local deviations from these smooth and perfectly sinusoidal trajectories. This in turns contributes to the identification of rhythmic but non-sinusoidal transcripts. Furthermore, we accommodate conditional dependence across probes through a latent factor framework. Dimensionality reduction and sparsity are induced through careful modeling of the latent factors as well as the local and Fourier basis coefficients. The proposed approach gives a comprehensive and easily-understood description of cyclic rhythms through posterior summaries of the model parameters, thus being of practical utility to biologists. In addition to the study in Jouffe et al. (2013), we apply our approach to the analysis of a microarray experiment on the plant model organism *Arabidopsis thaliana*. The original analysis and exposition of these experiments, together with a discussion of their biological significance is given in Jouffe et al. (2013) and Edwards et al. (2006), respectively.

1.3 Statistical analysis of non-homogeneous computer models

Large scale computer simulation is widely used in modern scientific research to investigate physical phenomena that are too expensive or impossible to replicate directly (Schade and Emanuel, 1999; Fan et al., 2009; Textor et al., 2009). Most simulators depend on a handful of tuning parameters and initial conditions, referred to as the input arguments. Often interest focuses on quantifying how uncertainty in the input arguments propagates through the simulator and produce a distribution function over one or many outputs of interest. In this paper we consider only deterministic simulators which when run on the same input twice will produce identical output values.

Quantifying uncertainty propagation will require several runs of a simulator at different input points to learn the input-output map $Y = f(\boldsymbol{x})$ accurately over the entire input space. However, computer simulations are very time-consuming, thus running a simulator over a dense grid of input points could be prohibitively expensive. On the other hand, running a simulator over a sparse design chosen in advance may result in insufficient information in vast parts of the input space. Consequently, there is considerable interest in estimating a slow computer simulator with a fast statistical emulator (Sacks et al., 1989; Kennedy and O'Hagan, 2001; Santner et al., 2003). The emulator is fitted to input-output data $\{\boldsymbol{x}^t, f^t\}$, where $f^t = \{f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_t)\}$ is obtained from a few preliminary runs of the simulator on design $\boldsymbol{x}^t = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$, and the fitted model is then used for prediction of f at input configurations not included in $\boldsymbol{x}^t$ (Sacks et al., 1989; Busby, 2009).

For Bayesian emulation, a common practice is to assign f a Gaussian process prior (Sacks et al., 1989; Currin et al., 1991; Schmidt and O'Hagan, 2000). Gaussian process emulation is appealing due to its mathematical tractability and ability to incorporate a wide range of smoothness assumptions. The conditional posterior distribution of f at future inputs, given data $\{\boldsymbol{x}^t, f^t\}$ and process hyperparameters, remains a Gaussian process distribution. The posterior mean of $f(\boldsymbol{x})$ gives a statistical estimate or surrogate for the simulator output at a new input $\boldsymbol{x}$, whereas the posterior variance at $f(\boldsymbol{x})$ quantifies how well the simulator has been learned at and around $\boldsymbol{x}$. The latter is a particularly attractive feature of Gaussian process emulation as it provides a model based assessment of the emulator's accuracy and could be used to actively learn an optimal sequence of input points on which the simulator needs to be run to minimize the uncertainty in posterior surface estimation.

Research on computer emulation has largely focused on stationary Gaussian process models (Sacks et al., 1989; Kennedy and O'Hagan, 2001). Stationary Gaussian processes regard the similarity between $f(\boldsymbol{x})$ and $f(\boldsymbol{x} + \boldsymbol{h})$ as a decaying function in $\boldsymbol{h}$ only, known up to global smoothness and decay parameters. This is a strong prior assumption that is not easily washed away by data and may lead to unrealistic emulation for many physical phenomena. In practice, stationary Gaussian process emulators run into difficulties when the shape of f has sharp localized features, e.g. abrupt discontinuities or tall peaks, and lead to poor point predictions and selection of future inputs. A simple example is illustrated in the left panel of Figure 1.1. Three aspects emerge: (i) the discovery of a tall peak in the middle has a rippling effect and creates large oscillations of the predictive mean curve over a large part of the input space, a phenomenon often called spline tension effect in the predictor form; (ii) prediction seems overconfident around the peak, where the error bars are too narrow to capture the high variability around $x = 0$; instead, (iii) prediction intervals are quite large where abrupt changes in the function values are not observed, and f is relatively more well-behaved. A sequential design strategy based on uncertainty quantified by the prediction variance would favor the selection of a new input from the whole x domain, with the only exception of the tall peak. Thus, stationary Gaussian processes favor the selection of new points in unexplored regions of the input space (exploration), but tend to neglect regions that are deemed important based on the current estimate of f (exploitation).

Extrinsic diagnostics is often used to assess the adequacy of a Gaussian process emulator as surrogate for the simulator (Bayarri et al., 2007; Bastos and O'Hagan, 2009). For example, one can examine the leave-one-out cross validated standardized residuals to quantify the emulator's uncertainty. Either too large or very small cross validated standardized residuals (as compared to a $N(0, 1)$ or a t_ν) at some validating points indicate that the emulator is poorly estimating the predictive uncertainty. Outliers of this kind denote a local fitting problem, which could be improved upon by adding new points in the vicinity. Thus, cross validation examines the local behavior of f, and flags those sub-regions where the simulator has more variations. Therefore, cross validation leans toward an exploitation-driven sequential design. Although cross validation is often combined with a stationary Gaussian process to better address sequential design, it is difficult to reconcile the exploration-driven predictive

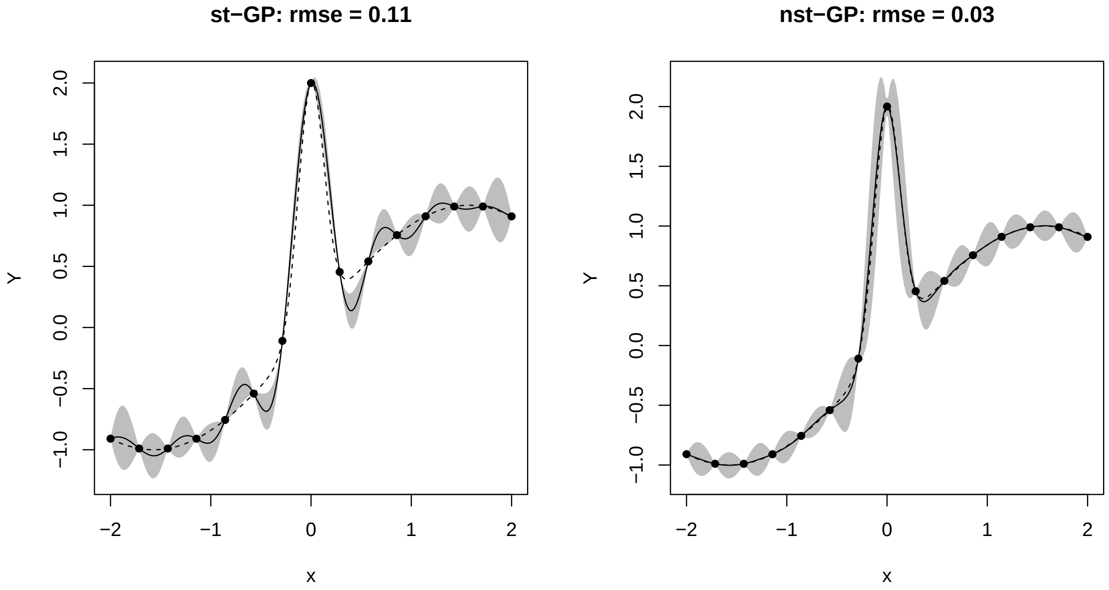


FIGURE 1.1: Plot of (true) function $f(x) = \sin(x) + 2\exp(-30x^2), x \in [-2, 2]$ (dashed line). The black dots represent observed data at 15 equally-spaced values of x. Left panel: the solid line is the point predictor of f, or conditional mean, obtained from a stationary Gaussian process emulator (st-GP) fitted to the data. Shaded areas represent the error bars. Right panel: non-stationary Gaussian process (nst-GP) via latent input augmentation. The root mean squared error (RMSE) is also reported.

variance of a stationary Gaussian process with the exploitation-driven flagging of cross validation, and any combination is ad-hoc. Also, the model remains misspecified: a stationary model is used for a response which is often intrinsically not so (Busby, 2009).

Several approaches to the problem of how to specifying non-stationary Gaussian process models can be found in the literature. In the context of computer emulation, Gramacy and Lee (2008) propose the Bayesian treed Gaussian process model, which applies independent stationary Gaussian processes to subregions of the input space determined by data-driven recursive partitioning parallel to the coordinate axes. Because of the parallel partitioning, treed Gaussian process adapts well to surfaces having rectangular local features (axes-aligned non-stationarity). However, it may run into difficulties when the nature of the non-stationarity is more general. Also, treed Gaussian process' hard partitioning of the input space prevents borrowing of information across partitions and enforces discontinuity on the estimated response surface. Ba and Joseph (2012) decompose f into the sum of two stationary Gaussian processes, the first capturing the smooth global trend and the second modeling local details. Other approaches in the context of Gaussian process regression include Sampson and Guttorp (1992); Schmidt and O'Hagan (2000); Paciorek and Schervish (2004). This literature makes it clear that the main challenges in non-stationary Gaussian process modeling are to keep the number of hyperparameters under control to facilitate efficient learning from limited data while allowing for non-stationary features of various geometric shapes and at the same time not to enforce non-stationarity when not needed.

We propose a non-stationary Gaussian process emulator (Section 5.1.2) by equipping a stationary Gaussian process with optional non-stationarity. Specifically, non-stationarity is achieved by augmenting the input space with one extra latent input which we infer from the data. The latent input can flag regions of the input space characterized by abrupt changes of the function values and help correct for inadequacies in the fit. In the example above we find the proposed method to give significantly improved performance (Figure 1.1). Furthermore, several numerical examples (Section 5.3) show that our emulator adapts to local features of many kinds of shape and provides a more trustworthy judgement of uncertainty than stationary and other existing non-stationary Gaussian process emulators. The latter is a key advantage of our method. When an emulator is used to actively learn an optimal sequence of design points to minimize expensive runs of the simulator, it is absolutely crucial to have trustworthy judgement of uncertainty of the current estimate of f to concentrate efforts only on where needed. Sections 5.4 and 5.5 show results from various synthetic and real experiments where a sequential version of our emulator outperforms similar sequential adaptations of existing Gaussian process emulators, where we measured performance by number of simulator runs needed to achieve a certain accuracy.

The proposed method is also attractive from an operational point of view. Both the latent input dimension and the response function (of the original plus the latent inputs) are individually modeled as stationary Gaussian processes controlled by a small number of hyperparameters that can be efficiently learned with sequential Monte Carlo computing leveraging on conjugacy properties of Gaussian process. Sequential Monte Carlo computing seamlessly blends with active learning of the sequential design, as opposed to Markov chain sampling based non-stationary Gaussian process emulators whose sequential adaptation requires re-running the whole Markov chain sampler at every iteration. We also investigate a two stage fast approximation of the proposed emulator where the latent input Gaussian process is directly learned from data through nonparametric regression and the estimated input surface is plugged in to learn f. Simulations suggest that the two stage approximation performs at least as well as the sequential Monte Carlo full Bayes counterpart in handling local features and selecting additional inputs from the boundaries of such features. However, the sequential Monte Carlo version of our emulator often achieves better accuracy given the same number of input points in the design.

2

Bayesian latent factor regression for functional and longitudinal data

In studies involving functional data, it is commonly of interest to model the impact of predictors on the distribution of the curves, allowing flexible effects on not only the mean curve but also the distribution about the mean. Characterizing the curve for each subject as a linear combination of a high-dimensional set of potential basis functions, we place a sparse latent factor regression model on the basis coefficients. We induce basis selection by choosing a shrinkage prior that allows many of the loadings to be close to zero. The number of latent factors is treated as unknown through a highly-efficient, adaptive-blocked Gibbs sampler. Predictors are included on the latent variables level, while allowing different predictors to impact different latent factors. This model induces a framework for functional response regression in which the distribution of the curves is allowed to change flexibly with predictors. We assess the performance of our approach through simulation studies and the methods are applied to data on blood pressure trajectories during pregnancy.

2.1 Functional latent factor regression model

Let n denote the number of subjects in the study. We suppose that functional data on subject i are available as noisy measurements of an underlying smooth curve $f_i(t)$ at n_i time points t_{ij}, $j = 1, \cdots, n_i$. We denote these measurements as y_{ij} and model

$$y_{ij} = f_i(t_{ij}) + \epsilon_{ij}, \tag{2.1}$$

with $\epsilon_{ij} \sim N(0, \varphi^2)$, independently across i and j. In our motivating application (Chapter 3), y_{ij} denotes the blood pressure (BP) measurement of the i-th woman at her j-th visit to the clinic during pregnancy, with t_{ij} denoting time (in weeks) from the onset of pregnancy.

To ensure smoothness, $f_1(t), \cdots, f_n(t)$ are assumed to belong to the linear span of a smooth finite basis $\{b_1(t), \cdots, b_p(t)\}$:

$$f_i(t) = \sum_{i=1}^{p} \theta_{il} b_l(t). \tag{2.2}$$

It is important to use a sufficiently large p and to choose locally concentrated basis elements so that a rich variety of shapes for $f_i(t)$ are entertained. In particular, after standardizing the time domain to $[0, 1]$, we use Gaussian kernels

$$b_1(t) = 1, \quad \text{and} \quad b_{l+1}(t) = \exp(-\nu \|t - \psi_l\|^2), \quad l = 1, \cdots, p-1 \tag{2.3}$$

with equally spaced kernel locations $\psi_1, \cdots, \psi_{p-1}$ and a bandwidth parameter ν to be specified later. By denoting the functional data vector of subject i by $\mathbf{y}_i$, we can write

$$\mathbf{y}_i = \mathbf{B}_i \boldsymbol{\theta}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim \mathrm{N}_{n_i}(0, \varphi^2 \mathbf{I}) \tag{2.4}$$

where $\mathbf{B}_i$ is the $n_i \times p$ matrix with rows $\{b_1(t_{ij}), \cdots, b_p(t_{ij})\}$, $j = 1, \cdots, n_i$ and $\boldsymbol{\theta}_i = (\theta_{i1}, \cdots, \theta_{ip})'$.

The coefficient vectors $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n$ capture all subject-to-subject variations in the functional data. But these vectors are non-sparse. They have a large dimension p and have highly correlated neighboring elements unless $f_1(t), \cdots, f_n(t)$ are sparse in the basis $\{b_l\}$. The latter is unlikely to hold for a pre-specified local basis such as ours. The non-sparsity of $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n$ makes them unfit to be included in a joint model with other observations of interest.

We obtain an attractive low dimensional representation of the curves by placing a sparse latent factor model on the basis coefficients

$$\boldsymbol{\theta}_i = \boldsymbol{\Lambda}\boldsymbol{\eta}_i + \boldsymbol{\zeta}_i, \quad \text{with} \;\; \boldsymbol{\zeta}_i \sim \mathrm{N}_p(0, \boldsymbol{\Sigma}) \tag{2.5}$$

where $\boldsymbol{\Lambda} = ((\lambda_{lm}))$ is a $p \times k$ factor loading matrix with $k \ll p$, $\boldsymbol{\eta}_i = (\eta_{i1}, \cdots, \eta_{ik})'$ is a vector of latent factors for subject i and $\boldsymbol{\zeta}_i = (\zeta_{i1}, \cdots, \zeta_{ip})'$ is a residual vector that is independent with the other variables in the model and is normally distributed with mean zero and a diagonal covariance matrix $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \cdots, \sigma_p^2)$.

The low dimensional vectors $\boldsymbol{\eta}_1, \cdots, \boldsymbol{\eta}_n$ are used in all subsequent parts of our model where we seek to link the curves $f_1(t), \cdots, f_n(t)$ with other variables of interest. Like $\boldsymbol{\theta}_i$, the vector $\boldsymbol{\eta}_i$ can also be interpreted as a coefficient vector for subject i because we can write

$$f_i(t) = \sum_{m=1}^{k} \eta_{im}\tilde{\phi}_m(t) + r_i(t) \tag{2.6}$$

where $\tilde{\phi}_m(t) = \sum_{l=1}^{p} \lambda_{lm} b_l(t)$, $m = 1, \cdots, k$ form an unknown non-local basis to be learned from data and $r_i(t) = \sum_{l=1}^{p} \zeta_{il} b_l(t)$ is a function-valued random intercept. This decomposition, without $r_i(t)$, is analogous to an FPCA representation of $f_i(t)$, except that the latter requires the basis functions $\tilde{\phi}_1(t), \cdots, \tilde{\phi}_k(t)$ to be mutually orthogonal eigenfunctions. Although orthogonality enhances interpretability of the elements in the decomposition, this is not a primary concern in our application since we view the latent factorization only as a vehicle to link functional observations with other variables. To highlight this difference with FPCA, we refer to $\{\tilde{\phi}_m\}$ as a dictionary.

The size k and the elements of the dictionary $\{\tilde{\phi}_m\}$ depend on how $\mathbf{\Lambda}$ is modeled. We assign $\mathbf{\Lambda}$ a multiplicative, gamma process shrinkage (MGPS) (Bhattacharya and Dunson, 2011a) prior which favors an unknown but small dictionary size k (refer to Section 2.2 for details on the MGPS prior).

Given the sparsity of the data, it becomes mandatory to borrow information across the population of curves to improve inferences and predictions. Specifically, the LFRM model allows borrowing strength across the different subjects in estimating their functions in that the low dimensional dictionary functions $\{\tilde{\phi}_m\}$, their number, and the random intercept $r_i(t)$ are learnt by pooling information from all subjects.

The score vectors $\boldsymbol{\eta}_1, \cdots, \boldsymbol{\eta}_n$ can be put in any flexible joint model with other variables of interest. For example, information from a covariate $\mathbf{x}_i$ can be incorporated through a simple linear model

$$\boldsymbol{\eta}_i = \boldsymbol{\beta}'\mathbf{x}_i + \mathbf{\Delta}_i, \quad \mathbf{\Delta}_i \sim \mathrm{N}_k(0, \mathbf{I}) \tag{2.7}$$

where $\boldsymbol{\beta}$ is a $r \times k$ matrix of unknown coefficients, and with r denoting the dimension of $\mathbf{x}_i$. With a semi-conjugate model on $\boldsymbol{\beta}$, this specification leads to very efficient posterior computation via Gibbs updating, as we describe in the next sub-section. Despite the simplicity of this linear model, the resulting model on $f_1(t), \cdots, f_n(t)$ allows a very flexible accommodation of the covariate information. Conditionally on $(\{b_l\}_{l=1}^p, \mathbf{\Lambda}, \mathbf{\Sigma}, \boldsymbol{\beta}, \{\mathbf{x}_i\}_{i=1}^n)$, these curves are independent (finite rank) Gaussian processes with covariate dependent mean functions $\mathbb{E}[f_i(t)] = \sum_{m=1}^k \boldsymbol{\beta}_m' \mathbf{x}_i \tilde{\phi}_m(t)$ and a common covariance function $\mathbb{C}\mathrm{ov}\{f_i(t), f_i(s)\} = \sum_{m=1}^k \tilde{\phi}_k(t)\tilde{\phi}_m(s) + \sum_{l=1}^p \sigma_l^2 b_l(t) b_l(s)$, where $\boldsymbol{\beta}_m$ denotes the m-th column of $\boldsymbol{\beta}$.

2.2 Prior elicitation

A Bayesian formulation of our sparse LFRM is completed with priors for the parameters in (2.1)-(2.7). Given the dimensionality, it is practically important to choose conditionally conjugate priors that lead to efficient posterior computation via blocked Gibbs sampling. Typical priors for factor analysis constrain $\mathbf{\Lambda}$ to be lower triangular with positive diagonal entries using normal and truncated normal priors for the free elements of $\mathbf{\Lambda}$ and gamma priors for the residual precisions (Arminger, 1998; Lopes and West, 2004). However, following Bhattacharya and Dunson (2011a) we note that such constraints are unnecessary and unappealing in leading to order dependence and computational inefficiencies. Hence, we follow their lead in using a MGPS prior for the loadings as follows:

$$\lambda_{jh}|\phi_{jh},\tau_h \sim \mathrm{N}(0,\phi_{jh}^{-1}\tau_h^{-1}), \quad \phi_{jh} \sim \mathrm{Gamma}(\upsilon/2,\upsilon/2), \quad \tau_h = \prod_{l=1}^{h}\delta_l \tag{2.8}$$

$$\delta_1 \sim \mathrm{Gamma}(a_1,1), \qquad \delta_l \sim \mathrm{Gamma}(a_2,1), \quad l \geqslant 2 \tag{2.9}$$

$j = 1,\ldots,p$, $h = 1,\ldots,k$, $\delta_l, l \geqslant 1$, are independent, τ_h is a global shrinkage parameter for the hth column and ϕ_{jh}'s are local shrinkage parameters for the elements in the hth column. Under a choice $a_2 > 1$, the τ_h's are stochastically increasing favoring more shrinkage as the column index increases. The choice of this shrinkage prior allows many of the loadings to be close to zero while avoiding factor splitting, thus inducing effective basis selection. The number of latent factors, k, is treated as unknown and tuned as the sampler progresses. Refer to Appendix A for a detailed discussion on the adaptive choice of k.

The prior structure under our model is completed by

$$\sigma_j^{-2} \sim \mathrm{Gamma}(a_\sigma,b_\sigma), \qquad \text{and} \qquad \varphi^{-2} \sim \mathrm{Gamma}(a_\varphi,b_\varphi) \tag{2.10}$$

with $j = 1,\ldots,p$. Furthermore, consider $\boldsymbol{\eta}'_{\cdot j} \sim \mathrm{N}(\tilde{\mathbf{X}}'\boldsymbol{\beta}_j,\mathrm{I}_n)$, where $\boldsymbol{\eta}'_{\cdot j}$ denotes the j-th column of the $n \times k$ transpose of the matrix of latent factors $\boldsymbol{\eta}$, $\boldsymbol{\beta}_j$ denotes the

j-th column of the $r \times k$ matrix of coefficients $\boldsymbol{\beta}$ and $\tilde{\mathbf{X}}'$ denotes the transpose of the matrix of predictors $\tilde{\mathbf{X}}$. Each row $i, i = 1, \ldots, n$, of $\tilde{\mathbf{X}}'$ corresponds to the vector of predictors for subject i, $\boldsymbol{x}_i' = (x_{i1}, \ldots, x_{ir})$. A Cauchy prior is induced on the matrix of coefficients $\boldsymbol{\beta}$ as follows

$$\boldsymbol{\beta}_j \sim \mathrm{N}(0, \mathrm{Diag}(\omega_{lj}^{-1})), \quad \omega_{lj} \sim \mathrm{Gamma}(1/2, 1/2), \quad j = 1, \ldots, k, \quad l = 1, \ldots, r. \tag{2.11}$$

2.3 MCMC algorithm & computational considerations

The posterior computation proceeds via a straightforward Gibbs sampler, and is similar to the Markov Chain Monte Carlo (MCMC) algorithm for the sparse Bayesian infinite factor model in Bhattacharya and Dunson (2011a). The sampler cycles through the following steps:

- *Update of* $\boldsymbol{\Lambda}$: Sample $\lambda_{jh}, \delta_1, \delta_h, \phi_{jh}$ from the following posteriors:

    1. Denote the jth row of $\boldsymbol{\Lambda}_{k*}$ (the loading matrix $\boldsymbol{\Lambda}$ truncated to $k^* << p$) by $\boldsymbol{\lambda}_j$; then the $\boldsymbol{\lambda}_j$'s have independent conditionally conjugate posteriors given by

    $$\pi(\boldsymbol{\lambda}_j \mid -) \sim N_{k*}((\mathbf{D}_j^{-1} + \sigma_j^{-2}\boldsymbol{\eta}'\boldsymbol{\eta})^{-1}\boldsymbol{\eta}'\sigma_j^{-2}\boldsymbol{\theta}^{(j)}, (\mathbf{D}_j^{-1} + \sigma_j^{-2}\boldsymbol{\eta}'\boldsymbol{\eta})^{-1})$$

    with $\mathbf{D}_j^{-1} = \mathrm{diag}(\phi_{j1}\tau_1, \ldots, \phi_{jk}\tau_{k*})$, $\boldsymbol{\eta}' = [\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{k*}]$ and $\boldsymbol{\theta}^{(j)} = (\theta_{j1}, \ldots, \theta_{jn})$, for $j = 1, \ldots, p$.

    2. Sample ϕ_{jh} from

    $$\pi(\phi_{jh} \mid -) \sim \mathrm{Gamma}\left(\frac{v+1}{2}, \frac{v}{2} + \frac{\tau_h \lambda_{jh}^2}{2}\right)$$

    3. Sample δ_1 from

    $$\pi(\delta_1 \mid -) \sim \mathrm{Gamma}\left(a_1 + \frac{pk^*}{2}, 1 + \frac{1}{2}\sum_{l=h}^{k^*} \tau_l^{(1)} \sum_{j=1}^{p} \phi_{jl}\lambda_{jl}^2\right)$$

4. Sample δ_h from

$$\pi(\delta_h \mid -) \sim \text{Gamma}\left(a_2 + \frac{p^*}{2}(k-h+1), 1 + \frac{1}{2}\sum_{l=1}^{k^*}\tau_l^{(h)}\sum_{j=1}^{p}\phi_{jl}\lambda_{jl}^2\right)$$

for $h \geqslant 2$, where $\tau_l^{(h)} = \prod_{t=1,t\neq h}^{l}\delta_t$ for $h = 1,\ldots,p$.

The sampling begins with a very conservative choice of k^*, which is then automatically selected within the adaptive Gibbs sampler as described in Bhattacharya and Dunson (2011).

- *Update of σ_j^2:* Denoting as σ_j^{-2} the diagonal elements of $\mathbf{\Sigma}^{-1}$, sample σ_j^{-2}, $j = 1,\ldots,p$, from conditionally independent posteriors

$$\pi(\sigma_j^{-2} \mid -) \sim \text{Gamma}\left(\frac{n}{2} + a_\sigma, b_\sigma + \frac{\sum_{i=1}^{n}(\boldsymbol{\theta}_i - \mathbf{\Lambda}\boldsymbol{\eta}_i)^2}{2}\right)$$

- *Update of φ^{-2}:* Sample φ^{-2} from

$$\pi(\varphi^{-2} \mid -) \sim \text{Gamma}\left(\frac{N}{2} + a_\varphi, b_\varphi + \frac{\sum_{j=1}^{N}(y_j - \mathbf{\Theta}_j)^2}{2}\right)$$

where N denotes the total number of observations, $\mathbf{y}$ is a column vector which stacks the measurements for all women, $\mathbf{y} = (y_{1,t_{1,1}},\ldots,y_{n,t_{n,n_n}})'$, and $\mathbf{\Theta}$ is a $N \times 1$ column vector which stacks the scores for all subjects, $\mathbf{\Theta} = \{\mathbf{B}_i\boldsymbol{\theta}_i,\ldots,\mathbf{B}_n\boldsymbol{\theta}_n\}'$, where each $\mathbf{B}_i\boldsymbol{\theta}_i$ has dimension $n_i \times 1$ with n_i the number of measurements for subject i.

- *Update of $\boldsymbol{\beta}$ and ω elements:*

1. Given the prior $\omega_{lj} \sim \text{Gamma}(1/2, 1/2)$, $l = 1,\ldots,r$ and $j = 1,\ldots,k$, sample ω_{lj} from the full conditional posterior

$$\pi(\omega_{lj} \mid -) \sim \text{Gamma}\left(1, \frac{1}{2}\left(1 + \beta_{lj}^2\right)\right)$$

2. Sample the jth column of the matrix of coefficients $\boldsymbol{\beta}$ from the full conditional posterior

$$\pi(\boldsymbol{\beta}_j \mid -) \sim N\left(\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' + \mathbf{E}^{-1}\right)^{-1} \tilde{\mathbf{X}}\boldsymbol{\eta}'_{\cdot j}, \left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}' + \mathbf{E}^{-1}\right)^{-1}\right)$$

with matrix $\mathbf{E}$ corresponding to $\mathbf{E} = \mathrm{Diag}(\omega_{lj}^{-1})$, $l = 1, \ldots, r$ and $j = 1, \ldots, k$.

- *Update of $\boldsymbol{\eta}_i$:* Marginalizing out $\boldsymbol{\theta}_i$, the model can be rewritten as

$$\begin{aligned}\mathbf{y}_i &= \mathbf{B}_i\boldsymbol{\Lambda}\boldsymbol{\eta}_i + \mathbf{B}_i\boldsymbol{\zeta}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim N(0, \varphi^2\mathbf{I}_{n_i}), \quad \boldsymbol{\zeta}_i \sim N_p(0, \boldsymbol{\Sigma}) \\ &= \mathbf{B}_i\boldsymbol{\Lambda}\boldsymbol{\eta}_i + \boldsymbol{\alpha}_i^*, \quad \boldsymbol{\alpha}_i^* \sim N(0, \varphi^2\boldsymbol{I}_{n_i} + \mathbf{B}_i\boldsymbol{\Sigma}\mathbf{B}'_i)\end{aligned}$$

Thus, sample $\boldsymbol{\eta}_i$ from the full conditional posterior

$$\begin{aligned}\pi(\boldsymbol{\eta}_i \mid -) &\sim N(\mathbf{A}^{-1} \times \mathbf{C}, \mathbf{A}^{-1}) \\ \mathbf{A} &= \boldsymbol{\Lambda}'\mathbf{B}'_i(\varphi^2\boldsymbol{I}_{n_i} + \mathbf{B}_i\boldsymbol{\Sigma}\mathbf{B}'_i)^{-1}\mathbf{B}_i\boldsymbol{\Lambda} + \boldsymbol{I}_k \\ \mathbf{B} &= \boldsymbol{\beta}'\boldsymbol{x}_i + \boldsymbol{\Lambda}'\mathbf{B}'_i(\varphi^2\mathrm{I}_{n_i} + \mathbf{B}_i\boldsymbol{\Sigma}\mathbf{B}'_i)^{-1}\mathbf{y}_i\end{aligned}$$

- *Update of $\boldsymbol{\theta}_i$:* Sample $\boldsymbol{\theta}_i$ from conditionally independent posteriors

$$\begin{aligned}\pi(\boldsymbol{\theta}_i \mid -) \sim N_p((&\varphi^{-2}\mathbf{B}'_i\mathbf{B}_i + \boldsymbol{\Sigma}^{-1})^{-1}(\varphi^{-2}\mathbf{B}'_i\mathbf{y}_i + \boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}\boldsymbol{\eta}_i), \\ &(\varphi^{-2}\mathbf{B}'_i\mathbf{B}_i + \boldsymbol{\Sigma}^{-1})^{-1})\end{aligned}$$

A crucial aspect of our research is to ensure computational tractability to scale well in dimension and sample size. Our model builds more parametric (mostly linear) relationships between the different components, and the basis expansion chosen to represent the functions $\boldsymbol{f}_i$ induces posterior computation which involves the update of single, low dimensional component pieces. Thus, our structure leads to an efficient Gibbs sampler having block updating steps, while avoiding the need to invert large matrices. For example, the HPHB study (Chapter 3) contains data for 1,027 women with an average number of 10 measurements per subject (range $= [1, 25]$), for a total of $N = 10,290$ observations, and with 12 clinical predictors collected for each woman. The posterior update took 71 seconds per hundred iterations in Matlab on an Intel(R) Core(TM)2 Duo machine. Our approach scales well both in the number of subjects and number of measurements, with simulation experiments showing that cases with $n \approx 4,000$ and $N \approx 40,000$ can be accommodated (a few minutes required per hundred iterations), while larger experiments face serious time and memory constraints.

Preliminary sensitivity analyses will be required to adjust the priors and other model parameters to provide the best fit to the data. To save on computing time, it might be preferable to run the preliminary analyses on a randomly chosen subset of subjects and proceed to the analysis of the complete data set when one is satisfied with the choice of the hyperparameters and other parameter values. This choice is discussed in Appendix A.

2.4 Joint modeling extension for the HPHB study

It is of interest to extend our LFRM to allow joint modeling of a functional predictor with scalar responses. For example, there is substantial interest in relating the BP trajectories to gestational age (GA) at delivery, birth weight (BW), and preeclampsia (hypertension and proteinuria at time of delivery).

We start with a simple probit extension of our model to predict premature delivery. A bivariate probit model for preeclampsia and low birth weight (LBW = weight under 2500 grams) is outlined in Section 2.4.1, and a joint model for BW, GA and mean arterial blood pressure (MAP = 2/3 diastolic pressure + 1/3 systolic pressure) is presented in Section 2.4.3. These extensions involve straightforward modifications of the MCMC algorithm for the LFRM (Section 2.3), which includes additional steps to sample from the full conditional posterior distributions of the new model parameters.

2.4.1 Probit model for risk of preterm birth

Preterm birth refers to the birth of a baby of less than 37 weeks GA. Let $z_i^{pb} = 1$ if preterm birth and $z_i^{pb} = 0$ if full-term birth. We let $\mathrm{P}(z_i^{pb} = 1|\alpha, \boldsymbol{\gamma}, \boldsymbol{\eta}_i) = \Phi(\alpha + \boldsymbol{\gamma}'\boldsymbol{\eta}_i)$, where $\Phi(\cdot)$ denotes the standard normal distribution function. α is an intercept with a $\mathrm{N}(\Phi^{-1}(0.123), 0.25)$ prior, where the hyperprior mean is chosen to correspond to the national average of 12.3% in 2008 (Hamilton et al., 2010), $\boldsymbol{\eta}_i$ are the latent factors for subject i, and $\boldsymbol{\gamma}$ is a vector of unknown regression coefficients with prior distribution $\boldsymbol{\gamma} \sim \mathrm{N}_k(\boldsymbol{\mu}_\gamma, \boldsymbol{\Sigma}_\gamma)$.

The full conditional posterior distributions needed for Gibbs sampling are not automatically available, but we can rely on the data augmentation algorithm of Albert and Chib (1993) to facilitate the computation:

$$z_i^{pb} = \mathbb{1}(W_i > 0) \qquad \text{with} \qquad W_i \sim \mathrm{N}(\alpha + \boldsymbol{\gamma}'\boldsymbol{\eta}_i, 1)$$

so that $\mathrm{P}(z_i^{pb} = 1|\alpha, \boldsymbol{\gamma}, \boldsymbol{\eta}_i) = \Phi(\alpha + \boldsymbol{\gamma}'\boldsymbol{\eta}_i)$ by marginalizing out W_i. Therefore, the same set of latent factors impacts on the functional predictor via the basis coefficients $\boldsymbol{\theta}_i$ and on the response variables via the probability of preterm birth.

2.4.2 Bivariate probit model for preeclampsia and low birth weight

We develop a bivariate probit model to study the relationship between preeclampsia, LBW and gestational MAP. The sample proportion of LBW is 12%, thus slightly higher than the corresponding national rate of 8.2% in 2008 (Hamilton et al., 2010), whereas the sample proportion of preeclamptic women is 16%, far above the incidence of preeclampsia which typically affects 5-8% of all pregnancies (Cunningham et al., 2010).

Let us denote the outcome variables for preeclampsia and LBW as z_p^i and z_{lbw}^i, respectively. In particular, z_p^i is an indicator variable equal to 1 if woman i develops preeclampsia, and z^i_{lbw} is an indicator variable equal to 1 if woman i delivers a LBW infant.

We adopt a data augmentation approach and introduce two underlying normal variables, W^i_p and W^i_{lbw}, such that $z^i_p = \mathbb{1}(W^i_p > 0)$ and $z^i_{lbw} = \mathbb{1}(W^i_{lbw} > 0)$, with $(W^i_p, W^i_{lbw})' \sim \mathrm{N}(\boldsymbol{\mu}, \tilde{\boldsymbol{\Sigma}})$, and $\boldsymbol{\mu} = (\alpha_1 + \boldsymbol{\gamma}_1'\boldsymbol{\eta}_i, \alpha_2 + \boldsymbol{\gamma}_2'\boldsymbol{\eta}_i)'$ and $\tilde{\boldsymbol{\Sigma}} = \left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)$, with ρ controlling the dependence between z^i_p and z^i_{lbw}. The joint probability of preeclampsia and LBW is obtained by double integration of the bivariate normal distribution of the latent variables W^i_p and W^i_{lbw}

$$\Pr(z^i_p = 1, z^i_{lbw} = 1) = \int_0^\infty \int_0^\infty \mathrm{N}_2(W^i_p, W^i_{lbw}; \boldsymbol{\mu}, \tilde{\boldsymbol{\Sigma}}) dW^i_p, dW^i_{lbw}$$

Analogously, we can compute the marginal probability of observing preeclampsia and the marginal probability of LBW.

The Bayesian specification of the bivariate probit model is completed by choosing conditionally conjugate (normal and multivariate normal) prior distributions for the additional parameters. This choice is discussed in Appendix A.

Heterogeneity across subjects and dependence between the smooth function, $\boldsymbol{f}_i$, and the outcomes, z^i_p and z^i_{lbw}, is accommodated through the latent factors, $\boldsymbol{\eta}_i$, which impact on the MAP measurements via the basis coefficients $\boldsymbol{\theta}_i$ and on the probabilities of preeclampsia and LBW via the latent normal variables W^i_p and W^i_{lbw}.

Our goal is to compare sequential predictions of the probability of preeclampsia and LBW for a test sample of women at different times during gestation, say at weeks 20, 25, and so on. Predictions are expected to improve over time, and we aim to assess whether we can make a detection with some certainty sufficiently early during gestation or if it is necessary to wait until close to delivery to make an accurate prediction.

2.4.3 Joint model of birth weight, gestational age at delivery and blood pressure

Let $\mathbf{z}_i$ denote the outcome for subject i, $\mathbf{z}_i = (z_{ib}, z_{ig})$, with z_{ib} denoting the BW and z_{ig} the GA at delivery. To flexibly joint model GA at delivery and BW, we consider a two-component mixture-model of bivariate normal distributions

$$(z_{ig}, z_{ib}) \sim \sum_{h=0}^{1} \pi_{ih} \mathrm{N}(\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h) \tag{2.12}$$

This model can be equivalently specified as

$$(z_{ig}, z_{ib}) \sim \mathrm{N}(\boldsymbol{\mu}_{T_i}, \boldsymbol{\Sigma}_{T_i}) \qquad \text{with} \qquad T_i = \mathbb{1}(W_i > 0) \tag{2.13}$$

where $T_i \in \{0, 1\}$ is a latent variable indicating which class (z_{ig}, z_{ib}) belong to, and $\pi_{ih} = \mathrm{P}(T_i = h)$. We now let the W_i's have independent t-distributions using a scale mixture of normals construction:

$$W_i \sim \mathrm{N}\left(\alpha + \boldsymbol{\gamma}'\boldsymbol{\eta}_i, \tilde{\sigma}^2 \hat{\phi}_i^{-1}\right), \qquad \text{with} \qquad \hat{\phi}_i \sim \mathrm{Gamma}(\tilde{\nu}/2, \tilde{\nu}/2) \tag{2.14}$$

where $\boldsymbol{\gamma}$ a $k \times 1$ vector of unknown regression coefficients with normal prior distribution, $\boldsymbol{\gamma} \sim \mathrm{N}_k(\boldsymbol{\mu}^*_\gamma, \boldsymbol{\Sigma}^*_\gamma)$, $\boldsymbol{\eta}_i$ are the latent factors for subject i and $\alpha \sim \mathrm{N}(\Phi^{-1}(0.1), 0.25)$. Note that (2.14) constitutes a t approximation to a logit link function on the mixing weights π_{ih}, and to ensure a good approximation to the univariate logistic distribution we set $\tilde{\sigma}^2 \equiv \pi^2(\tilde{\nu} - 2)/3\tilde{\nu}, \tilde{\nu} \equiv 7.3$ (O'Brien and Dunson, 2004). In addition, this approximation ensures conjugacy of the full conditional distributions, thus allowing efficient posterior update. To complete our Bayesian specification, we chose an inverse-Wishart (I-W) distribution for the covariance matrix, $\boldsymbol{\Sigma}_h \sim \text{I-W}_2(\nu^*_h, \mathbf{V}_h)$, and a bivariate normal distribution for the mean $\boldsymbol{\mu}_h$, $\boldsymbol{\mu}_h \sim \mathrm{N}_2(\boldsymbol{\mu}^h_0, \boldsymbol{\Sigma}^h_{\mu 0})$. The choice of the hypeparameter values is discussed in Appendix A.

Therefore, the common set of latent factors impacts both on the functional predictor $\boldsymbol{f}_i$ and on the outcomes $\mathbf{z}_i = (z_{ig}, z_{ib})$ via the class membership probability of the pregnancy outcomes, $\pi_{i1}(\boldsymbol{\eta}_i) = \mathrm{P}(T_i = 1) = \Phi\left(\frac{\alpha + \boldsymbol{\gamma}'\boldsymbol{\eta}_i}{\sqrt{\tilde{\sigma}^2 \hat{\phi}_i^{-1}}}\right)$.

2.5 Simulation study

To evaluate the performance of our model and to compare it with related methods, we considered a simulation example. To make the simulated data more realistic and interpretable we based them on the Healthy Pregnancy, Healthy Baby (HPHB) study, assuming $n = 200$ and with the true parameters set equal to the posterior means from the real data analysis (Chapter 3). We generated samples of gestational age (in weeks) and birth weight (in Kg) from a two-component mixture of bivariate normal distributions with true means set equal to $\boldsymbol{\mu}_1 = (34.54, 2.27)'$ and $\boldsymbol{\mu}_2 = (38.17, 3.50)'$ and covariance matrices

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 1.516 & 0.261 \\ 0.261 & 1.235 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 1.212 & 0.185 \\ 0.185 & 1.221 \end{pmatrix}$$

We standardized time to the $[0, 1]$ interval, $t_{ij} \in [0, 1]$, and set $b_1(t_{ij}) = 1$ and $b_{l+1}(t_{ij}) = \exp\{-4||t_{ij} - \psi_l||^2\}, l = 1, \ldots, 9$, with ψ_l's equally spaced kernel locations in $[0, 1]$ and $p = 10$.

To implement our Bayesian analysis, we chose a Gamma$(0.5, 0.25)$ prior distribution with mean 2 for the diagonal elements of $\boldsymbol{\Sigma}^{-1}$, and we placed a Gamma(0.5, 0.2) with mean 2.5 on φ^{-2}. The gamma hyperparameter for ϕ_{jh} was set to be $\upsilon = 5$, $a_1 = a_2 = 1.5$ in (2.8)-(2.9) and a Cauchy prior was induced on the matrix of coefficients $\boldsymbol{\beta}$ (2.11). We chose $k = 4$ as the starting number of factors, and we adapted k according to the procedure described in Bhattacharya and Dunson (2011a). The MCMC algorithm was run for 25,000 iterations including a 5,000 iterations burn-in, and collected every 5th sample to thin the chain and reduce the autocorrelation in the posterior samples. Based on the examination of traceplots of function values at a variety of time locations and for different subjects, the sampler appeared to converge rapidly and to mix efficiently.

The average of the estimated number of factors was 11.37 corresponding to

$k_{\text{true}} = 11$, and with empirical 95% credible interval given by $[9, 13]$. The estimated posterior mean of $\boldsymbol{\mu}_1$ was (34.37, 2.35) and the estimated posterior mean for $\boldsymbol{\mu}_2$ was $(37.91, 3.42)$ respectively, with corresponding 95% credible intervals containing the true values of $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. The estimates of the covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ were

$$\hat{\boldsymbol{\Sigma}}_1 = \begin{pmatrix} 1.413 & 0.429 \\ 0.429 & 1.105 \end{pmatrix} \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}_2 = \begin{pmatrix} 1.098 & 0.265 \\ 0.265 & 1.152 \end{pmatrix}$$

with 95% credible intervals containing the true values of $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$.

The left panels of Figure 2.1 show the data, true curves and estimates under the LFRM for three randomly selected subjects. In general, estimates are very close to the true curves even when data are sparse, as for subject 122, and the true curves are always enclosed in the credible bounds.

We then obtained a smooth estimator of the covariance operator and its corresponding eigenfunctions as described by Crainiceanu and Goldsmith (2010). In contrast to the LFRM, FPCA does not allow to learn about the representation size k, thus we need to estimate the dimension of the functional space. As a fast alternative to cross-validation, we decided to retain a number of eigenfunctions such that the cumulative percentage of explained variance was greater than 90% and the explained variance by any single subsequent component was less than 5%. Therefore, we retained the first $k = 4$ eigenfunctions and obtained $\boldsymbol{\Lambda}$ as the least squares estimate of

$$\boldsymbol{\Psi} = \mathbf{B}^* \times \boldsymbol{\Lambda} \tag{2.15}$$

with $\boldsymbol{\Psi}$ denoting here the matrix of eigenfunctions and $\mathbf{O}_i \times \mathbf{B}^* = \mathbf{B}_i$, $\mathbf{B}_i$ denoting the design matrix for subject i and $\mathbf{O}_i$ representing an $(n_i \times T)$ matrix with column j equal to a column of 1's if subject i was measured at time j, $j = 1, \ldots, T$ (T denotes the number of unique time locations). We then repeated the analysis fitting the LFRM with $\boldsymbol{\Lambda}$ and the number of factors $k = 4$ fixed. We will denote this

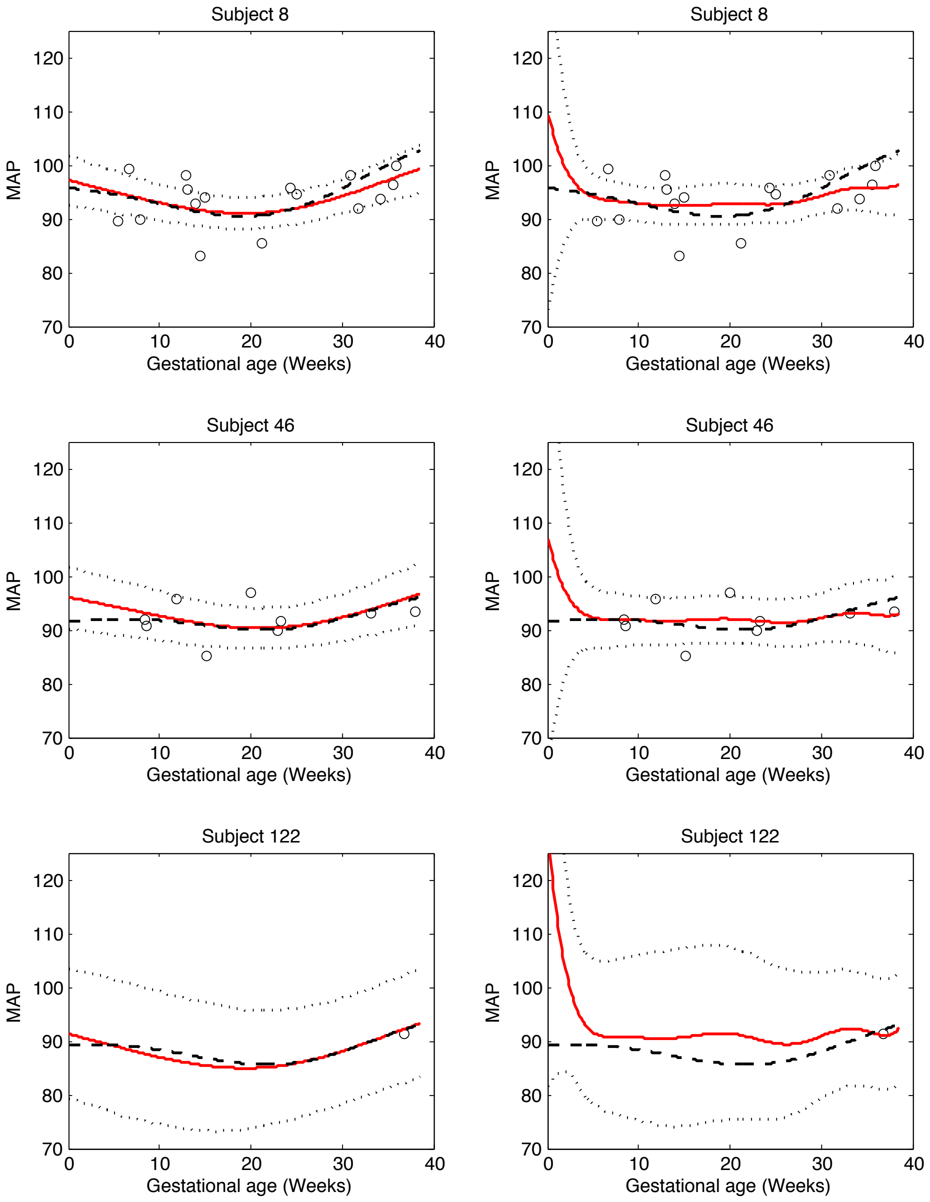


Figure 2.1: Data and function estimates for 3 subjects in the simulation example under the LFRM (left panels) and two-stage FPCA (right panels). The true functions are represented with dashed lines, the posterior means are solid lines, and the dotted lines are 95% pointwise credible intervals.

procedure as two-stage FPCA approach. Estimates are shown in the right panels of Figure 2.1. We can notice some deviations of the estimated curves from the true curves along the course of the entire pregnancy, with very wide confidence intervals at early pregnancy when typically no or few measurements are observed and when data are more sparse, as for subject 122. Notice also that for subject 8 the true curve is no longer enclosed within the credible bounds at delivery. The analysis was repeated retaining $k_{true} = 11$ eigenfunctions, but this did not lead to any significant improvement in the performance.

Under the two-stage FPCA approach, the estimated posterior mean of $\boldsymbol{\mu}_1$ was $(34.26, 2.31)$ and the estimated posterior mean for $\boldsymbol{\mu}_2$ was $(37.81, 3.39)$ respectively, with corresponding 95% credible intervals containing the true values of $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. The estimates of the covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ were

$$\hat{\boldsymbol{\Sigma}}_1 = \begin{pmatrix} 1.249 & 0.383 \\ 0.383 & 1.087 \end{pmatrix} \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}_2 = \begin{pmatrix} 1.227 & 0.305 \\ 0.305 & 1.165 \end{pmatrix}$$

To assess the predictive performance, we repeated the analysis holding out and predicting the MAP measurements collected after the $30th$ week of gestation for 100 randomly selected women having at least 1 observation in the first 30 weeks and 1 observation after the $30th$ week. Also, we fitted "baseline" LFRM and two-stage FPCA approach with no covariates setting $\boldsymbol{\eta}_i \sim \mathrm{N}(\mathbf{0}, \mathbf{I}_k)$. Results are reported in Table 2.1 together with the computing time in seconds per hundred of iterations. The high values of the predictive errors are not surprising given the presence of many outliers in the MAP measurements that are hard to predict. The LFRM leads to better predictive performance than the two-stage FPCA approach both with and without covariates. However, we notice that the predictive errors do not decrease with the incorporation of covariate information: this seems to suggest that in our blood pressure application the outcomes are predominantly learned from the random deviations rather than from the covariates. In fact, the MAP measurements are

Table 2.1: Mean square predictive error, predictive average absolute bias (PAAB) and predictive maximum absolute bias (PMAB) for the simulated data with the LFRM and the two-stage FPCA approach fitted with and without covariates, respectively. The computing time in seconds is per hundred of iterations.

	LFRM		two-stage FPCA	
	Covariates	No Covariates	Covariates	No Covariate
MSPE	70.02	69.31	73.45	74.11
PAAB	6.63	6.61	6.80	6.83
PMAB	27.74	27.50	28.63	28.72
Comp. Time	40	21	34	15

affected by great variability that makes them hard to predict despite the available covariate information. Figure 2.2 shows the estimated joint distribution of gestational age (in weeks) and birth weight (in Kg) for subjects 8 and 46 in the simulation example under the LFRM, along with corresponding contour plots. The true values of gestational age at delivery and birth weight correspond to (38.82, 3.47) and (33.51, 1.41) for subject 8 and subject 46, respectively. The joint distribution is bimodal, with the two components of the Gaussian mixture clearly distinct, and with the joint model assigning higher mass to the true component each subject belongs to, that is, the second component for subject 8 and the first component for subject 46. The posterior probability of being in component 1 is 0.3057 for subject 8, and increases to 0.6025 for subject 46. Analogous results are obtained with the two-stage FPCA approach, with posterior probabilities of being in component 1 being equal to 0.2938 and 0.5592 for subjects 8 and 46, respectively.

The analysis was repeated under different choices of the hyperparameter values and initial number of factors for the LFRM. The results were robust, with no noticeable differences in the conclusions.

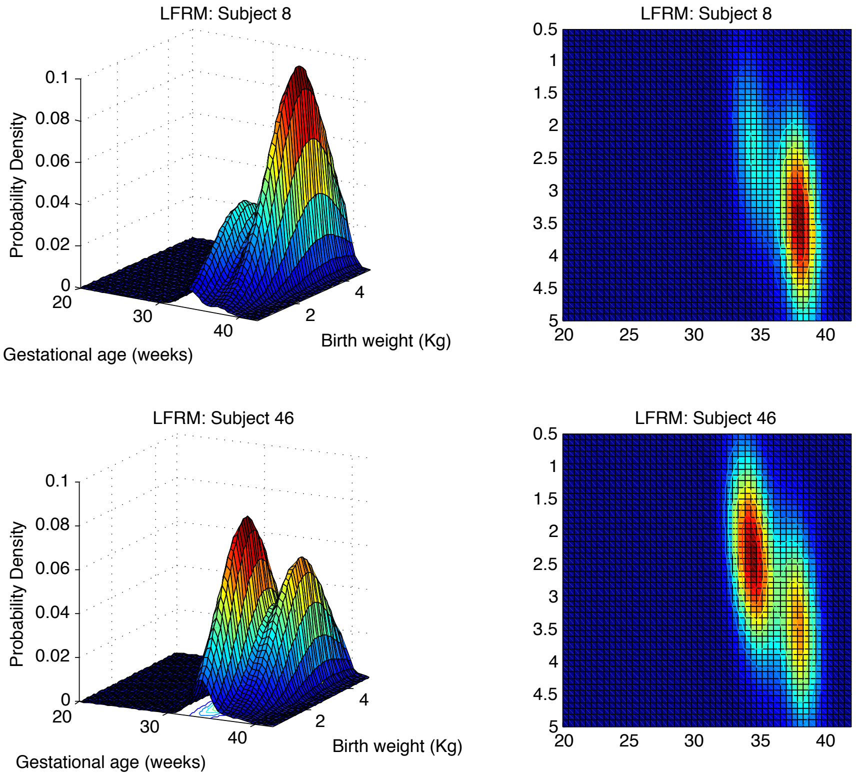


FIGURE 2.2: LFRM-estimated joint distribution of gestational age (weeks) and birth weight (Kg) and contour plot for subjects 8 and 46 in the simulation example.

3

Application to the HPHB study

3.1 Overview

The applied context that partially motivated some of the methodological and computational research presented in Chapter 2 concerns the so-called Healthy Pregnancy, Healthy Baby study (HPHB), an ongoing prospective cohort study examining the effects of environmental, social, and host factors on racial disparities in pregnancy outcomes. The HPHB study is part of the US EPA-funded Southern Center on Environmentally Driven Disparities in Birth Outcomes and enrolls pregnant women from the Duke Obstetrics Clinic and the Durham County Health Department Prenatal Clinic. Our focus is on the investigation of gestational mean arterial blood pressure ($\text{MAP} = 2/3$ diastolic pressure $+ 1/3$ systolic pressure). It is well known that hypertensive women are more likely to experience complications during pregnancy than normotensive women (Cunningham et al., 2010). In particular, gestational hypertension is associated with low birth weight (LBW) and early delivery, and in the most serious cases the mother develops preeclampsia. In normotensive women, blood pressure (BP) typically declines steadily until mid-gestation and then rises until delivery.

In contrast, preeclamptic women typically experience no early decline in BP, with BP remaining stable during the first half of pregnancy and then rising until delivery. Also, primiparous, older, and non-Hispanic black women are more likely than other demographic groups to experience hypertensive disorders during pregnancy. Monitoring the gestational BP can help identify women at risk of adverse birth outcomes, and point to appropriate treatments.

Data were available for 1,027 English-literate women at least 18 years old, for a total of 10,290 measurements. Women with twin gestation or with known congenital anomalies were not included in our analysis. Women with pre-gestational chronic hypertension were also excluded since their BP was artificially lowered by medical treatment. Moreover, we only considered non-Hispanic black and non-Hispanic white women due to the limited number of Hispanics and other ethnic groups in the study.

3.2 Analysis and results

The sampler described in Section 2.3 was run for 25,000 iterations, with the first 5,000 samples discarded as a burn-in and collecting every fifth sample to thin the chain. The sampler appeared to converge rapidly and mix efficiently based on the examination of traceplots of function estimates $f_i(t_{ij})$ at a variety of time locations and for different subjects. The estimated number of factors was 11, with a 95% credible interval of [9, 13].

Figure 3.1 shows the results for 6 randomly selected women, with the MAP estimates following the typical U-shaped trajectory. Repeating the analysis for the two-stage FPCA approach (Figure A.1), we observe accurate estimates at locations close to data points, but the estimates are inferior when no or few measurements are recorded. The use of a pre-specified, over-complete set of basis functions with no shrinkage on $\boldsymbol{\Lambda}$ (and hence no basis selection) leads to overly-spiky curves.

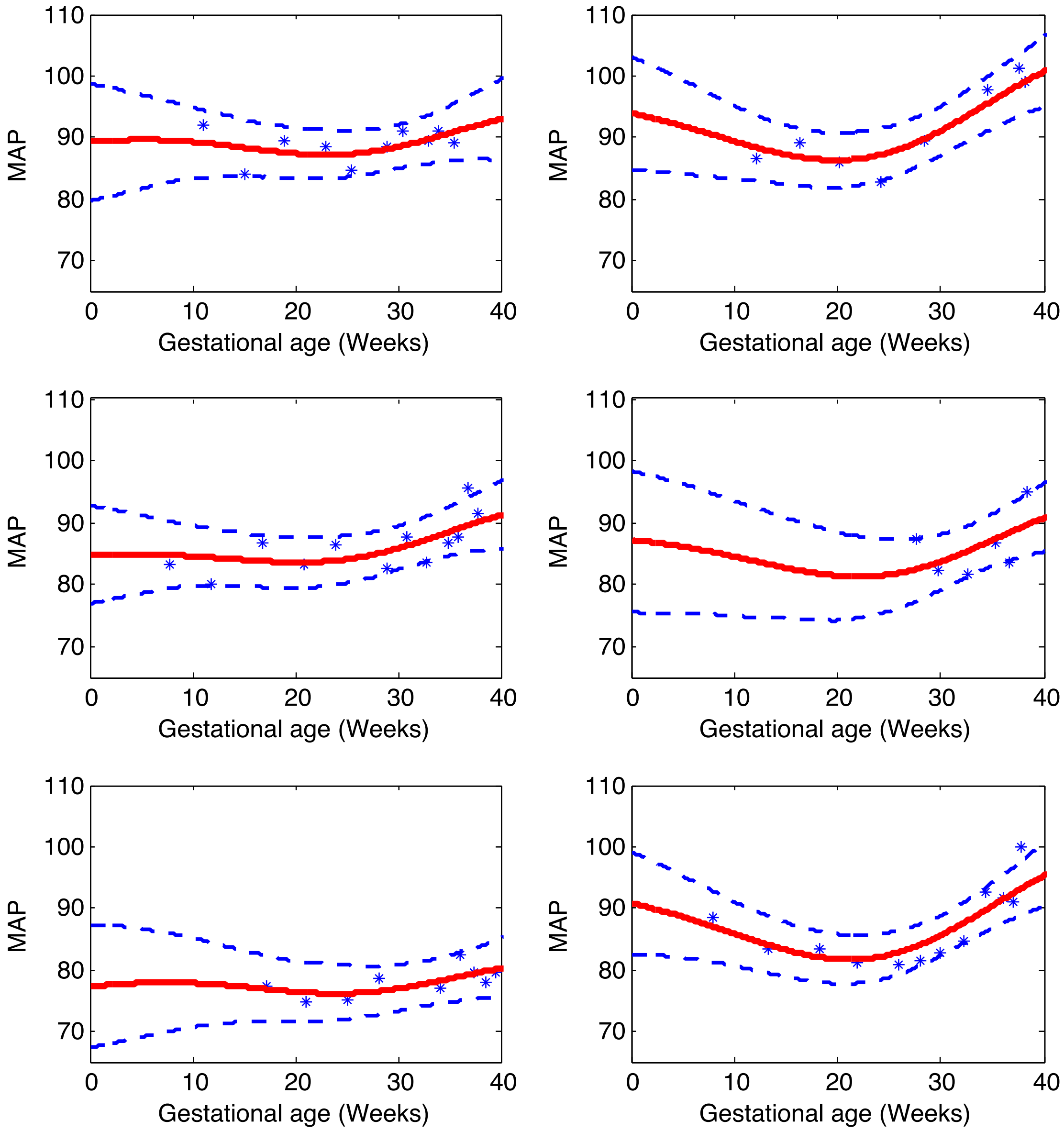


FIGURE 3.1: MAP function estimates for 6 randomly selected women in the Healthy Pregnancy, Healthy Baby Study. The posterior means are solid lines and dashed lines are 95% pointwise credible intervals. The x-axis scale is time in weeks starting at the estimated day of ovulation.

To assess the predictive performance, we held out and predicted the MAP measurements collected after the $30-th$ week for 300 randomly selected women with at least one measurement in the first 30 weeks. We then compared our approach with

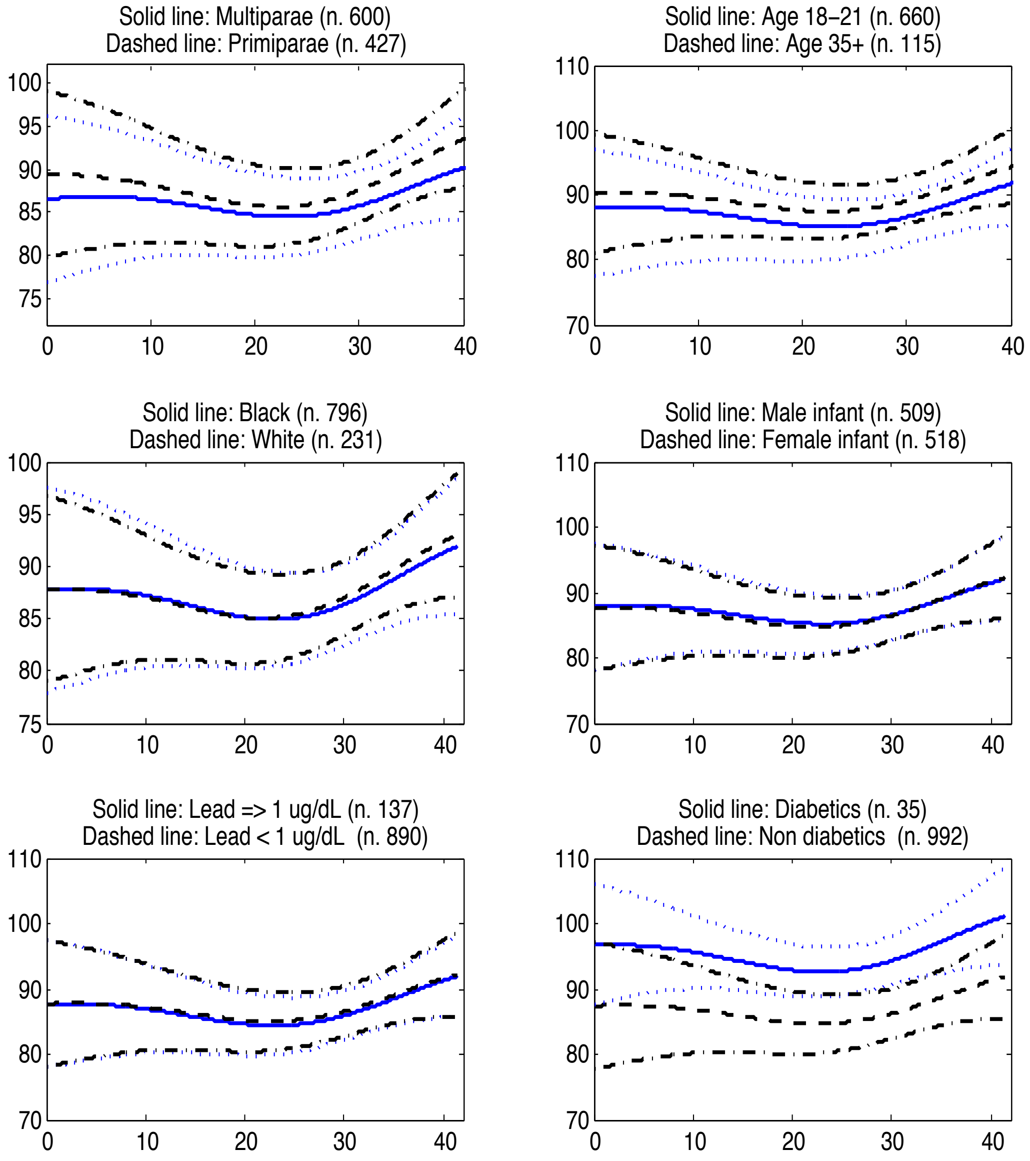
Table 3.1: Mean square predictive error (MSPE), predictive average absolute bias (PAAB) and predictive maximum absolute bias (PMAB) for the HPHB study with the LFRM and the two-stage FPCA approach fitted with and without covariates, respectively.

	LFRM		Two-stage FPCA	
	Covariates	No Covariates	Covariates	No Covariate
MSPE	88.36	89.91	92.16	92.22
PAAB	7.44	7.51	7.52	7.52
PMAB	43.50	43.29	49.51	49.62

"baseline" LFRM and two-stage FPCA with no covariates by setting $\boldsymbol{\eta}_i \sim \mathrm{N}(\mathbf{0}, \mathbf{I}_k)$. Results are reported in Table 3.1. The high prediction errors were expected since there were many hard-to-predict outliers in the MAP measurements. Predictions improved with the LFRM, although the inclusion of covariate information did not significantly decrease the prediction errors.

Figure 3.2, which shows how average MAP trajectories change across six different covariate groups, confirms previous findings on gestational BP, with older and primiparous women having higher BP, although discrepancies are small. Diabetic women have higher gestational BP than healthy women, with non-overlapping 95% credible intervals between mid-gestation and the $35th$ week. There were no differences among the remaining covariate groups.

To assess the relative importance of the j-th covariate, we look at the j-th column of the $k \times r$ matrix $\boldsymbol{\beta}'$, which contains the vector of coefficients associated with covariate j. The norms of the columns of $\boldsymbol{\beta}'$ indicate whether the covariates have any impact on the latent factors. The magnitude of the elements within each column determines the load of the covariate on each latent factor. If $||\boldsymbol{\beta}'_{\cdot j}|| = 0$, covariate j does not impact on the estimate of any of the latent factors for any subject. Figure 3.3 shows side-by-side boxplots of the norms of the posterior estimates of the columns


Solid line: Multiparae (n. 600)
Dashed line: Primiparae (n. 427)

Solid line: Age 18−21 (n. 660)
Dashed line: Age 35+ (n. 115)

Solid line: Black (n. 796)
Dashed line: White (n. 231)

Solid line: Male infant (n. 509)
Dashed line: Female infant (n. 518)

Solid line: Lead => 1 ug/dL (n. 137)
Dashed line: Lead < 1 ug/dL (n. 890)

Solid line: Diabetics (n. 35)
Dashed line: Non diabetics (n. 992)

FIGURE 3.2: MAP function estimates for 6 representative covariate groups. The dotted line represents the 95% pointwise credible interval of the blue solid line; the dash-dot line represents the 95% pointwise credible interval and refers to the black dashed line.

of $\boldsymbol{\beta}'$. Greater relative impact is attributed to the indicators for renal disease and (age > 35), followed by lead and cadmium concentration in ng/mL and maternal

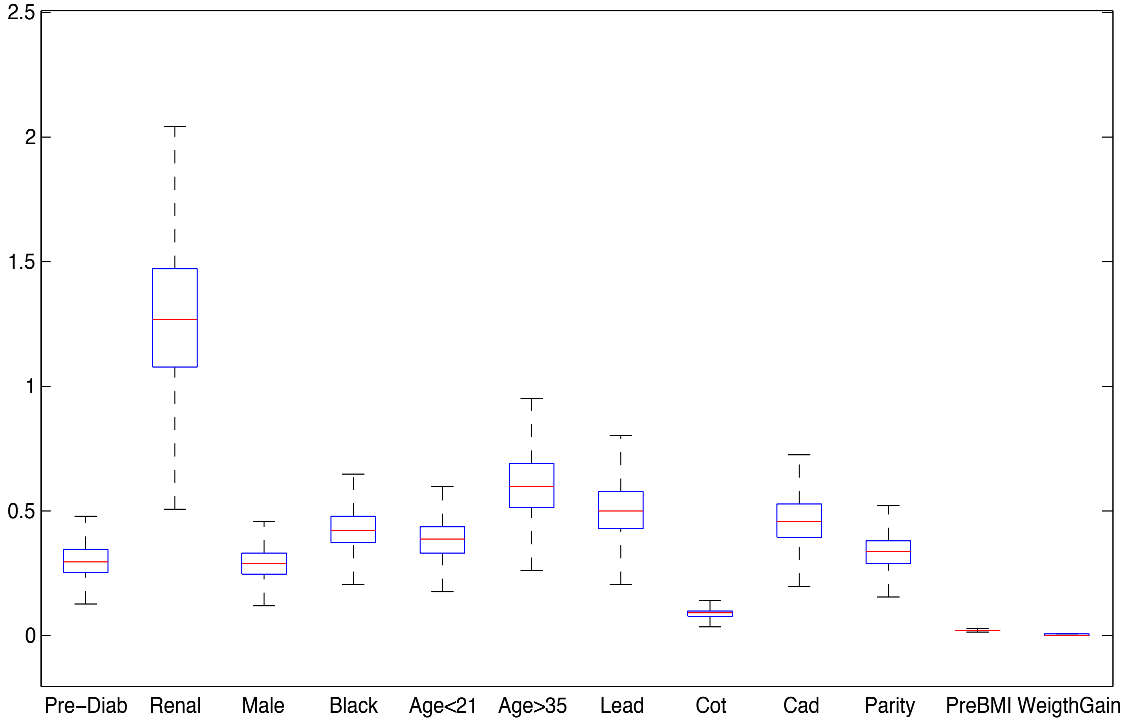


FIGURE 3.3: Side-by-side boxplots of the norms of the posterior estimates of the columns of $\boldsymbol{\beta}$.

race. Similarly, one can look at the norms of the columns of $\boldsymbol{\Lambda}$ to assess the relative impact of $\boldsymbol{\Lambda\eta}_i$ on $\boldsymbol{\theta}_i$ (Figure A.2).

In terms of joint modeling, we report the results of a probit extension used to predict LBW. For this analysis, we randomly split the data into a training set of 677 women and a test set of 350 women. The complete data was retained for the training set whereas the test set was entirely held out, that is, neither the MAP measurements nor the final outcome were included. We compared the LFRM with the Dependent Dirichlet process (DDP) in De la Cruz-Mesia et al. (2007), the Kernel partition process (KPP) in Dunson (2010), and with two-stage FPCA. The ROC plot in Figure 3.4 shows that the LFRM outperforms the two-stage FPCA approach, and it is equally good as KPP in guaranteeing high sensitivity. However, the LFRM's

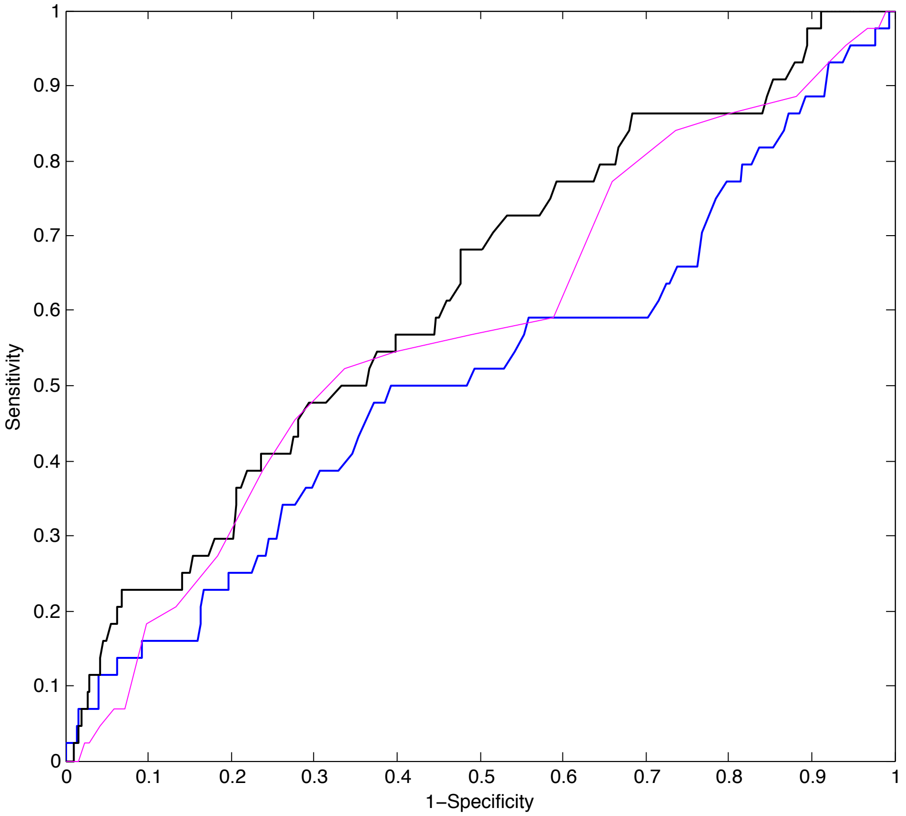


FIGURE 3.4: ROC plot for the correct classification of LBW in the HPHB study: the black line refers to the LFRM, the magenta line to the KPP, and the blue line to two-stage FPCA.

classification performance could be potentially improved over the KPP (which does not include covariates) by letting the predictors directly impact on the probability of LBW, while currently only an indirect impact via the $\boldsymbol{\eta}_i$'s is accommodated. The DDP had worse performance than our approach, so the ROC curve was omitted for simplicity of exposition.

Table 3.2 reports the posterior mean estimates of the marginal probabilities of preeclampsia and LBW (with Monte Carlo standard errors) computed at the $20th$, $25th$, $30th$ and $35th$ week of gestation for four randomly selected women in the test set. The final outcome information was included for women in the training

Table 3.2: Posterior mean estimates of the probabilities of preeclampsia and LBW (with Monte Carlo standard errors). z_p^i and z_{lbw}^i are indicator variables equal to 1 if woman i developed preeclampsia and delivered a LBW infant, respectively. Woman 1: $z_p^1 = 1$, $z_{lbw}^1 = 1$; Woman 2: $z_p^2 = 1$, $z_{lbw}^2 = 0$; Woman 3: $z_p^3 = 0$, $z_{lbw}^3 = 1$; Woman 4: $z_p^4 = 0$, $z_{lbw}^4 = 0$.

	Subjects			
$\Pr(z_p^i = 1)$	1	2	3	4
20*th* week	0.2545 (0.0037)	0.2085 (0.0034)	0.0711 (0.0019)	0.1179 (0.0025)
25*th* week	0.2819 (0.0047)	0.1314 (0.0031)	0.1148 (0.0038)	0.1046 (0.0027)
30*th* week	0.3640 (0.0044)	0.1960 (0.0035)	0.0855 (0.0023)	0.0985 (0.0023)
35*th* week	0.4185 (0.0042)	0.1141 (0.0023)	0.1128 (0.0024)	0.0983 (0.0021)
$\Pr(z_{lbw}^i = 1)$	1	2	3	4
20*th* week	0.2582 (0.0054)	0.0858 (0.0032)	0.2544 (0.0053)	0.1144 (0.0037)
25*th* week	0.2391 (0.0056)	0.0644 (0.0030)	0.3166 (0.0062)	0.0981 (0.0038)
30*th* week	0.3193 (0.0058)	0.0986 (0.0035)	0.2865 (0.0057)	0.1056 (0.0036)
35*th* week	0.3462 (0.0058)	0.0608 (0.0027)	0.3462 (0.0058)	0.0997 (0.0034)

set only, while the BP measurements at time of delivery were available for none of the women. Women in the test set had at least one MAP measurement before the 20*th* week, and at least one measurement after the 35*th* week. As early as 20 weeks of gestation, the LFRM estimated probabilities of preeclampsia and LBW were up to three times higher than the national rates for women who in fact experienced preeclampsia and/or LBW, with one exception being the probability of preeclampsia for woman 2, which was initially high but then dropped to 11.41% at the 35*th* week. By looking at Figure A.3, it is evident that the curve and the BP measurements for woman 2 were similar to those of normotensive woman 4. Thus, it is possible that woman 2 had normal BP during the prenatal visits, but was still preeclamptic because she had very high BP (and proteinuria) at delivery.

These findings suggest that, as early as the 20*th* week of gestation, the LFRM identifies women at high risk for adverse birth outcomes, with predictions getting more accurate around the $30th$ to $35th$ week of gestation. However, the LFRM may fail to identify the risk of preeclampsia in women who only register a sharp increase in MAP at delivery since the normotensive gestational BP would not be enough to detect the risk of the adverse outcome.

3.3 Discussion

We proposed a Bayesian latent factor regression model for functional data. The basic formulation generalizes the sparse Bayesian infinite factor model of Bhattacharya and Dunson (2011a), which was developed for estimation of high-dimensional covariance matrices for vector data, to the functional data case. This allows one to include a high-dimensional set of pre-specified basis functions, while allowing automatic shrinkage and effective removal of basis coefficients not needed to characterize any of the curves under study. The proposed framework has the advantage of straightforward computation via a simple Gibbs sampler. In addition, we consider several generalizations allowing predictors to impact on the latent factor scores and accommodating joint modeling of functional predictors with scalar responses that are modeled parametrically or via mixture models. Along the same lines, we can consider joint modeling of multiple related functions easily within the proposed framework, but our emphasis was on developing methods motivated by the application to the study of blood pressure and pregnancy outcomes.

4

Bayesian analysis of dependent functional data with application to circadian studies

The identification of circadian-regulated genes is a crucial step toward discovering physiological processes that are clock-controlled. Clock-genes are usually detected by searching for periodic time-course gene expression profiles in microarray data. However, common approaches do not accommodate for the potential dependence across genes. We develop a Bayesian methodology for periodicity identification that explicitly takes into account the complex correlation structure across time course trajectories in the gene expressions. We employ a latent factor representation to accommodate dependence and verify patterns and relationships between genes, while representing the true trajectories in the Fourier domain allows for inference on period, phase, and amplitude of the signal. We allow for the identification of circadian genes through a carefully chosen variable selection prior on the Fourier basis coefficients. Although motivated by time-course gene expression array data, the proposed methodology is applicable to the analysis of dependent functional data at broad.

4.1 Methodology

4.1.1 Overview

We consider data from a typical gene expression time course experiment in the form of a $p \times T$ matrix $\mathbf{Y} = \{y_{ij}\}$. Generic element y_{ij} denotes the observed messenger ribonucleic acid (mRNA) concentration for gene i at time t_j for $i = 1, \dots, p$, where p denotes the total number of genes. In circadian microarray studies, data are typically collected over two complete circadian cycles and the sampling rate is usually two of four hours depending on the particular experiment under investigation. In the mouse liver ribosomal proteins' expression study (Jouffe et al., 2013) the sampling rate is two hours, thus $t_j = 0, 2, 4 \dots, 46$ and $T = 24$. Hereafter, we will make explicit reference to the study in Jouffe et al. (2013), which motivates our methodological research, although the structure applies more generally to any circadian microarray experiment with the appropriate choices of T and sampling rate.

The observed signal for protein i consists of noisy measurements of the underlying smooth true profile at T time points. Thus, we observe

$$y_{ij} = f_i(t_j) + \nu_{ij} \tag{4.1}$$

where y_{ij} are error-prone measurements of the underlying true signal. Suppose that the de-trended and centered true signal for protein i at time t_j, $f_i(t_j)$, can be decomposed as

$$f_i(t_j) = \sum_{m=1}^{q} (\theta_{i,2m-1} b_{2m-1}(t_j) + \theta_{i,2m} b_{2m}(t_j)) = \boldsymbol{\theta}_{i,m}^\top \mathbf{b}_m(t_j) = \boldsymbol{\theta}_{i,m}^\top \mathbf{b}_{m,j},$$

where for $m = 1, \dots, q$ we define $\boldsymbol{\theta}_{i,m} = (\theta_{i,2m-1}, \theta_{i,2m})^\top$ and $\mathbf{b}_{m,j} = [b_{2m-1}(t_j), b_{2m}(t_j)]^\top$. The vector

$$\mathbf{b}_j = [b_1(t_j), b_2(t_j), \ldots, b_{2q-1}(t_j), b_{2q}(t_j)]^\top$$

represents a set of $2q$ fixed basis functions evaluated at time t_j. One popular basis for a space of periodic functions is the Fourier basis

$$\mathbf{b}(t) = \left[\sin\left(\frac{2\pi}{\omega_1}t\right), \cos\left(\frac{2\pi}{\omega_1}t\right), \ldots, \sin\left(\frac{2\pi}{\omega_q}t\right), \cos\left(\frac{2\pi}{\omega_q}t\right)\right],$$

where $\{\omega_m\}_{m=1}^q$ denotes the periodicity of the signal and t is time represented by a unit-interval increase. The q period lengths w_m are assumed known and fixed. Since there are 24 time points per transcript in the mouse liver ribosomal proteins dataset, we can use up to twelve sine/cosine pairs of harmonics. According to Nyquist-Shannon theorem and common sense, the possible range of periods would be $4, 6, 8, 10, 12, 14, 16, 18, 20, 22$ or 24 hours, but biologists would argue to reduce this list to $4, 6, 8, 12$ or 24 hours only. In practice, suitable period lengths can be proposed by inspecting the average periodogram of the probes and choosing the frequencies of the q ordered largest peaks in the spectrum. Here w_1 is the shortest period, and $w_2, \ldots, w_q$ correspond to longer periods.

The term ν_{ij} in Equation 4.1 models the deviation between the observed measurement at time t_j, y_{ij}, and the underlying smooth profile. In the original study (Jouffe et al., 2013), two mice are sacrificed every two hours and the reported p expression levels at time j are obtained by pooling 3 μg o total mRNA from each mice. Therefore, the ν_{ij}'s are correlated across probes, i. Specifically, each profile at time j may deviate from its own underlying truth because of a "mouse effect" which could make, e.g. a certain protein more expressed than the corresponding truth and another protein less expressed at time j. To accommodate dependence across probes at time j we adopt a sparse factor model:

$$\boldsymbol{\nu}_j = \boldsymbol{\Lambda}\boldsymbol{\eta}_j + \boldsymbol{\epsilon}_j, \tag{4.2}$$

with $\boldsymbol{\nu}_j = [\nu_{1j}, \ldots, \nu_{pj}]^\top$, $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p]^\top$ is a $p \times k$ factor loading matrix with elements $\{\lambda_{ih}\}_{i=1,\ldots,p;\ h=1,\ldots,k}$, $\boldsymbol{\eta}_j = (\eta_{1j}, \ldots, \eta_{kj})^\top$ is $k \times 1$ vector of latent factors at time j which explains mice-specific deviations of the expression levels at time j from their corresponding truth (it explains why proteins at time j may be systematically over- or under-expressed with respect to the "truth"), and $\boldsymbol{\epsilon}_j$ is a residual error. Sparsity here is necessary given the very large p, and Section 4.1.2 discusses how sparsity can be achieved through the modeling of $\boldsymbol{\Lambda}$.

The full model for subject i at time t_j is

$$\begin{aligned} y_{ij} &= g_i(t_j) + \boldsymbol{\lambda}_i^\top \boldsymbol{\eta}_j + \epsilon_{ij}, \quad \text{with} \quad \epsilon_{ij} \sim N(0, \sigma_i^2), \qquad (4.3) \\ g_i(t_j) &= f_i(t_j) + \boldsymbol{c}_j^\top \boldsymbol{\gamma}_i = \mathbf{b}_j^\top \boldsymbol{\theta}_i + \boldsymbol{c}_j^\top \boldsymbol{\gamma}_i, \end{aligned}$$

where the first term $\mathbf{b}_j^\top \boldsymbol{\theta}_i = \mathbf{b}(t_j)^\top \boldsymbol{\theta}_i$ captures periodic oscillations, the second term $\boldsymbol{c}_j^\top \boldsymbol{\gamma}_i = \boldsymbol{c}(t_j)^\top \boldsymbol{\gamma}_i$ captures local deviations from the underlying periodic oscillation (if present), and the third term $\boldsymbol{\lambda}_i^\top \boldsymbol{\eta}_j$ captures across-proteins dependence (if present). The importance of the $\boldsymbol{c}_j^\top \boldsymbol{\gamma}_i$ component becomes evident in studies where mice are given a stimulus at the beginning of the experiment. The stimulus might produce deviations of the observed expression levels from the true signals, and these deviations shall manifest at different times across proteins and last for a different amount of time, if present at all. After standardizing the time domain to $[0, 1]$, suitable choices for $\boldsymbol{c}(t_j)^\top$ are Gaussian kernels

$$c_l(t_j) = \exp\{-\psi ||t_j - \xi_l||^2\}, \qquad l = 1, \ldots, \tilde{T} \qquad (4.4)$$

with equally spaced kernel location $\xi_1, \ldots, \xi_{\tilde{T}}$ and bandwidth parameter ψ, or B-splines basis functions. $\boldsymbol{\theta}_i$ ($\boldsymbol{\gamma}_i$) is the $2q \times 1$ ($\tilde{T} \times 1$) vector of fixed periodic (local) basis function coefficients for protein i. Greater $\tilde{T}$ corresponds to more flexibility in modeling local deviations. We follow standard practice in normalizing the data prior to analysis and hence do not include an intercept term in (4.3). We use $\mathbf{y}_i =$

$(y_{i1},\ldots,y_{iT})^\top$ to denote the i-th row of $\mathbf{Y}$ (the i-th protein observed at times $1,\ldots,T$); and $\mathbf{y}^{(j)} = (y_{1j},\ldots,y_{pj})^\top$ to denote the j-th column of $\mathbf{Y}$ (p proteins observed at the time j). Construction (4.3) for y_{ij} can now be rewritten in vector notation as

$$\mathbf{y}_i \quad = \quad \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\nu}_i = \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i + \boldsymbol{\varepsilon}_i, \tag{4.5}$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1^\top \\ \vdots \\ \mathbf{b}_T^\top \end{bmatrix} \in \Re^{T\times 2q}; \;\; \mathbf{C} = \begin{bmatrix} \boldsymbol{c}_1^\top \\ \vdots \\ \boldsymbol{c}_T^\top \end{bmatrix} \in \Re^{T\times \tilde{T}}; \;\; \boldsymbol{\lambda}_i \in \Re^k; \;\; \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1{}^\top \\ \vdots \\ \boldsymbol{\eta}_T{}^\top \end{bmatrix} \in \Re^{T\times k};$$

$\boldsymbol{\varepsilon}_i = (\varepsilon_{i1},\ldots,\varepsilon_{iT})^\top \sim N_T(\mathbf{0},\sigma_i^2\boldsymbol{I}_T)$ with $T \times T$ identity matrix $\boldsymbol{I}_T$;

or

$$\mathbf{y}^{(j)} \quad = \quad \boldsymbol{\Theta}\mathbf{b}_j + \boldsymbol{\Gamma}\boldsymbol{c}_j + \boldsymbol{\nu}^{(j)} = \boldsymbol{\Theta}\mathbf{b}_j + \boldsymbol{\Gamma}\boldsymbol{c}_j + \boldsymbol{\Lambda}\boldsymbol{\eta}_j + \boldsymbol{\varepsilon}^{(j)}, \tag{4.6}$$

where

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1^\top \\ \vdots \\ \boldsymbol{\theta}_p^\top \end{bmatrix} \in \Re^{p\times 2q}; \;\; \boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\gamma}_1^\top \\ \vdots \\ \boldsymbol{\gamma}_p^\top \end{bmatrix} \in \Re^{p\times \tilde{T}}; \;\; \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1^\top \\ \vdots \\ \boldsymbol{\lambda}_p^\top \end{bmatrix} \in \Re^{p\times k};$$

$$\boldsymbol{\varepsilon}^{(j)} = (\varepsilon_{1j},\ldots,\varepsilon_{pj})^\top \sim N_p(\mathbf{0},\boldsymbol{\Sigma}),\; \boldsymbol{\Sigma} = \mathrm{diag}\{\sigma_1^2,\ldots,\sigma_p^2\}$$

The latent factors $\boldsymbol{\eta}_j$ have a natural interpretation as (mice-specific) unobserved traits of the two mice sacrificed at time j that explain the dependence structure across proteins at time t_j. Hereafter we follow standard practice and assign a normal prior to the latent factors at time t_j, $\boldsymbol{\eta}_j \sim N(\mathbf{0},\boldsymbol{I}_k)$. Proteins are assumed to be independent given the latent factors, and dependence among proteins is induced by marginalizing over the distribution of the factors, so marginally $\mathbf{y}^{(j)} \sim N(\boldsymbol{\Theta}\mathbf{b}_j + \boldsymbol{\Gamma}\boldsymbol{c}_j, \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Sigma})$. In practical applications involving moderate to large p, the number of factors k is typically much smaller than p, thus inducing a sparse characterization of the unknown covariance matrix $\mathbf{\Lambda}\mathbf{\Lambda}^\top + \mathbf{\Sigma}$.

The next Section discusses suitable prior choices for the model parameters in Equations (4.5)-(4.6) and examines how these choices translate into the ability to perform period identification.

4.1.2 Prior elicitation

With regard to the modeling of the basis coefficients $\{\boldsymbol{\theta}_i, \boldsymbol{\gamma}_i\}_{i=1}^p$, we need a technical device to induce sparsity / parsimony, hence avoid over-fitting, whilst retaining an easy interpretation of the method. The latter is particularly crucial for the modeling of $\boldsymbol{\theta}_i$ since inference on this set of parameters is the primary interest of our work. Although different formulations are possible, we adopt the latent threshold model (LTM) of Nakajima and West (2013). The LTM is a direct extension of standard Bayesian variable selection which assigns non-zero prior probabilities to zero values of regression parameters, and continuous priors centered at zero otherwise.

We begin introducing the LTM for the elements of $\boldsymbol{\gamma}_i$. Denote with γ_{il} the lth component of the $\tilde{T} \times 1$ vector of local basis coefficients $\boldsymbol{\gamma}_i$. The model assumes

$$\gamma_{i,l} = \tilde{\gamma}_{i,l}\mathbb{1}(|\tilde{\gamma}_{i,l}| \geqslant \varpi^*_{i,l}), \tag{4.7}$$

where $\varpi^*_{i,l} \geqslant 0$ is a latent threshold and $\mathbb{1}(\cdot)$ denotes the indicator function. Equation (4.7) embodies sparsity/shrinkage and parameter reduction when necessary, with the lth local basis coefficient shrunk to zero when it falls below a threshold. If the true smooth profile for protein i is given by the oscillatory behavior measured by $\mathbf{B}\boldsymbol{\theta}_i$ with no time localized deviations, then each component of vector $\tilde{\boldsymbol{\gamma}}_i = \{\tilde{\gamma}_{i,l}\}_{l=1}^{\tilde{T}}$ is expected to be uniquely shrunk to zero. Non-zero components allow for time-localized deviations. The vector $\tilde{\boldsymbol{\gamma}}_i$ is modeled as

$$\tilde{\boldsymbol{\gamma}}_i = \mathbf{Z}\boldsymbol{\lambda}_i + \boldsymbol{\alpha}_i^\gamma \qquad \text{and} \qquad \boldsymbol{\alpha}_i^\gamma \sim N_{\tilde{T}}(\mathbf{0}, \boldsymbol{I}), \tag{4.8}$$

where $\mathbf{Z}$ is a $\tilde{T} \times k$ matrix and $\boldsymbol{\lambda}_i$ is the vector of factor loadings for protein i as in (4.3). We assign a (multivariate) standard normal prior to the rows of $\mathbf{Z}$, $\mathbf{Z}_j^\top \sim N_k(\mathbf{0}, \boldsymbol{I})$, $j = 1, \ldots, \tilde{T}$.

We adopt the same variable selection prior for the periodic basis coefficients. Denote with $\boldsymbol{\theta}_{im} = \{\theta_{i,2m-1}, \theta_{i,2m}\}^\top$ the vector of $2m-1$th and $2m$th components of $\boldsymbol{\theta}_i$, $m = 1, \ldots, q$. Thus, $\theta_{i,2m-1}$ is the coefficient of the $2m-1$th sine basis and $\theta_{i,2m}$ is the coefficient of the $2m$th cosine basis, both harmonics of period w_m. To enhance a correct interpretation of periodicity, we need to switch off $\theta_{i,2m-1}$ and $\theta_{i,2m}$ jointly provided that shrinkage is supported by the data. Therefore, we assume:

$$\boldsymbol{\theta}_{i,m} = \tilde{\boldsymbol{\theta}}_{i,m} \mathbb{1}(||\tilde{\boldsymbol{\theta}}_{i,m}|| \geqslant \varpi_{i,m}), \tag{4.9}$$

where $\varpi_{i,m}$ is a latent threshold. The idea behind (4.9) is that the value of the w_m-periodic basis coefficients is shrunk to zero when their norm falls below a mth- (and protein-) specific threshold. Suppose, for example, that the set of probable periods is $\{4, 6, 8, 12, 24\}$ hours, i.e. $q = 5$. If the ith time series $\mathbf{y}_i = (y_{i1}, \ldots, y_{iT})^\top$ is generated with a true signal $f(t) = A \sin\left(\frac{2\pi}{12}t + \varphi\right)$, which is a 12 hours periodic function with phase φ hours and amplitude A, then column i of $\boldsymbol{\Theta}$ is expected to contain only two non-zero elements, $\theta_{i,7}$ and $\theta_{i,8}$, for representing sum of 12-hour periodic basis functions $b_7 = \sin\left(\frac{2\pi}{12}t\right), b_8 = \cos\left(\frac{2\pi}{12}t\right)$. There are two possibilities of parametrization for $\theta_{i,7}$ and $\theta_{i,8}$: first $\theta_{i,7} = A\sin(\varphi)$, $\theta_{i,8} = A\cos(\varphi)$; and second $\theta_{i,7} = A\cos(\varphi)$, $\theta_{i,8} = A\sin(\varphi)$. Clearly, this lack of identifiability in $\boldsymbol{\theta}_{i,4}$ does not affect inference for phase and amplitude.

Further, we assume $\tilde{\boldsymbol{\theta}}_i = \{\tilde{\boldsymbol{\theta}}_{i,m}\}_{m=1}^q$ is modeled as

$$\tilde{\boldsymbol{\theta}}_i = \mathbf{W}\boldsymbol{\lambda}_i + \boldsymbol{\alpha}_i^\theta \qquad \text{and} \qquad \boldsymbol{\alpha}_i^\theta \sim N_{2q}(\mathbf{0}, \boldsymbol{I}), \tag{4.10}$$

where $\mathbf{W}$ is a $2q \times k$ matrix and $\boldsymbol{\lambda}_i$ is the vector of factor loadings for protein i. Similar to the structure on $\mathbf{Z}$, we assume $\mathbf{W}_j^\top \sim N_k(\mathbf{0}, \boldsymbol{I})$, $j = 1, \ldots, 2q$. This simple structure on $\boldsymbol{\gamma}_i$ and $\boldsymbol{\theta}_{i,m}$ in (4.7)-(4.9) allows to flexibly take into account the dependence among parameters $\boldsymbol{\gamma}_i$, $\boldsymbol{\theta}_{i,m}$ and λ_i.

To continue, we adopt a multiplicative gamma process shrinkage prior (MGPSP) on the loadings

$$\begin{aligned}\lambda_{ih}|\phi_{ih},\tau_h &\sim \mathrm{N}(0,\phi_{ih}^{-1}\tau_h^{-1}), \quad \phi_{ih}\sim \mathrm{Ga}\left(\frac{\rho}{2},\frac{\rho}{2}\right), \quad \tau_h=\prod_{l=1}^{h}\zeta_h \\ \zeta_1 &\sim \mathrm{Ga}(a_1,1), \quad \zeta_l\sim \mathrm{Ga}(a_2,1), \quad l\geqslant 2, \quad i=1,\dots,p,\end{aligned}$$

with $h = 1, \dots, k$,. In matrix notation, row i of $\boldsymbol{\Lambda}$ has prior

$$\boldsymbol{\lambda}_i^\top|\{\phi_{ih}\}_{h=1}^k,\{\tau_h\}_{h=1}^k \sim N_k(\mathbf{0},\boldsymbol{D}_i), \tag{4.11}$$

with $\boldsymbol{D}_i = \mathrm{diag}(\phi_{i1}^{-1}\tau_1^{-1},\dots,\phi_{ik}^{-1}\tau_k^{-1})$. The MGPSP prior was introduced in Chapter 2. Refer to Section 2.2, Appendix A, and Bhattacharya and Dunson (2011b) for additional details.

To conclude the model formulation, we need to specify prior distributions on the latent threshold parameters. The straightforward extension of Nakajima and West (2013) to our scenario leads to a dependent prior for $\varpi_{i,m}$ (and $\varpi_{i,l}^*$) of the type $\varpi_{i,m} \sim \mathrm{Unif}(0, U_{i,m})$ for $i = 1, \dots, p$ and $m = 1, \dots, q$, where the upper bound of the uniform prior is function (thus dependent) of other model parameters. Nakajima and West (2013) give a thorough discussion on the choice of the upper bound $U_{i,m}$ and its impact on the sparsity structure of the model. We recognize, however, that a dependent prior on the latent thresholds would lead to unnecessary complications in the posterior update of some model parameters whitin our construction. Therefore, we opt for independent priors on the latent thresholds

$$\begin{aligned}
\varpi_{i,m} &\sim \text{Unif}(0, K_\theta), \quad i = 1, \ldots, p, \text{ and } m = 1, \ldots, m \qquad (4.12)\\
\varpi^*_{i,l} &\sim \text{Unif}(0, K_\gamma), \quad i = 1, \ldots, p, \text{ and } l = 1, \ldots, \tilde{T} \qquad (4.13)\\
K_\theta &\sim \text{Pareto}(a_\theta, b_\theta), \qquad (4.14)\\
K_\gamma &\sim \text{Pareto}(a_\gamma, b_\gamma). \qquad (4.15)
\end{aligned}$$

K_θ and K_γ are fundamental sparsity parameters shared across subjects. Smaller or larger degrees of expected sparsity might be needed depending on the context, thus these parameters need to be inferred from the data. In general, the smaller these parameters are estimated to be the less sparse the model becomes. Clearly, there is no inherent interest in direct inference on the thresholds themselves; the interest is their roles as defining the ability to shrink parameters when the data support sparsity. Correspondingly, there is no interest in the underlying values of the latent $\tilde{\boldsymbol{\gamma}}_i$ and $\tilde{\boldsymbol{\theta}}_i$ when below threshold.

In addition to sparsity, K_θ and K_γ play an important role in controlling for multiplicity adjustments. When analyzing microarray expression data, tens of thousands of genes are estimated simultaneously, so the problem of multiple testing must be considered. Müller et al. (2006) remark that posterior inference adjusts for multiplicities, and no further adjustment is required, provided that the probability model includes a positive prior probability of non-periodic expression for each protein i, and a hyperparameter that defines the prior probability mass for non-periodic expression. In our context, the two conditions are controlled and satisfied by treating K_θ and K_γ as model parameters with hyperpriors as in (4.14)-(4.15). A discussion on the role and specification of hyperparameters $a_\theta, b_\theta, a_\gamma, b_\gamma$ is deferred to the next Section.

4.1.3 Prior sparsity probabilities

The hyperparameters in Equations (4.14)-(4.15) have a key role in defining the prior (and posterior) probability of shrinkage of the periodic and local basis coefficients. Depending on the data and context, different degrees of sparsity might be desired, thus requiring careful tuning of these parameters. For simplicity, we focus here on the impact that different choices of a_γ and b_γ have on the prior probability of shrinkage of the local basis coefficients $\gamma_{i,l}$'s.

With no loss of generality, assume $\mathbf{Z}_l^\top \boldsymbol{\lambda}_i = 0$ so that $\tilde{\gamma}_{i,l} \mid \mathbf{Z}_l, \boldsymbol{\lambda}_i \sim N(0,1)$. Conditional on the latent thresholds, one can easily derive the probability that coefficient $\gamma_{i,l}$ is not switched off as

$$P(\gamma_{i,l} \neq 0 \mid \varpi_{i,l}^*) = 2\{1 - \Phi(\varpi_{i,l}^*)\}, \tag{4.16}$$

where $\Phi(\cdot)$ denotes the standard normal cumulative density function. By marginalizing over the prior distribution of the thresholds, we obtain

$$p_\gamma := P(\gamma_{i,l} \neq 0 \mid K_\gamma) = \int_0^{K_\gamma} 2\{1 - \Phi(\varpi_{i,l}^*)\} \times \frac{1}{K_\gamma} d\varpi_{i,l}^*, \tag{4.17}$$

which corresponds to the prior probability of non-shrinkage conditional on K_γ. Note that p_γ is neither protein-dependent (i) nor basis-dependent (l). Integral (4.17) is not available in closed form, but can be evaluated via numerical integration. If we denote with $\gamma_{i,l}^*$ the indicator for the decision of not shrinking parameter $\gamma_{i,l}$, then the $\gamma_{i,l}^*$'s are independent and identically distributed (*i.i.d*) random variables with common probability of success p_γ,

$$\gamma_{i,l}^* := \mathbb{1}(\gamma_{i,l} \neq 0) \mid p_\gamma \overset{i.i.d.}{\sim} \text{Bernoulli}(p_\gamma). \tag{4.18}$$

Though separate decisions of shrinkage are to be taken on each parameter $\gamma_{i,l}$, the different cases are treated in unison within a framework of exchangeability.

To evaluate the dependence of p_γ to hyperparameters a_γ and b_γ, we generated independent realizations of K_γ from the prior distribution (4.15) given a particular choice of a_γ and b_γ. For each of these realizations, we evaluated integral (4.17) and constructed the histogram of the so-obtained p_γ's. Figure 4.1 shows the distribution of p_γ for different choices of a_γ and b_γ, with b_γ varying along the rows and a_γ along the columns. It emerges clearly that b_γ defines an upper bound on the prior probability of non-shrinkage. Larger choices of b_γ determine a smaller upper bound on p_γ (the upper bound of the x-axis decreases by moving along the rows), thus effectively favoring a more sparse structure. For any given value of b_γ, small (large) values of a_γ tend to favor smaller (larger) p_γ whereas for $a_\gamma \approx 1$ the prior distribution of p_γ becomes

$$p_\gamma \overset{approx}{\sim} \text{Uniform}(0, U_{p_\gamma}), \tag{4.19}$$

where U_{p_γ} is the upper bound on the distribution of p_γ. This result is true for larger choices of b_γ in particular.

A context where we expect high sparsity with, say, 90% thresholding implies a fairly high choice of b_γ, and a value of $b_\gamma = 10$ or above leads to a marginal sparsity probability exceeding 0.92. Unless the context involves substantive information to suggest favoring smaller or larger degrees of expected sparsity, an approximately uniform prior with $a_\gamma = 1$ and $b_\gamma = 5$ or 10 is a good default for p_γ. A similar reasoning follows for p_θ.

4.1.4 Period detection

The LTM on the periodic basis coefficients eases the identification of those proteins that are more likely to be periodically expressed. Denote with TS the total number of thinned posterior samples post-burn-in obtained by running a Markov-Chain-Monte-Carlo (MCMC) algorithm to update the model parameters, i.e. $TS = \frac{\text{Tot. \# runs} - \text{burn-in}}{\text{thin}}$ (see Section 4.2 for details). We can easily derive the posterior

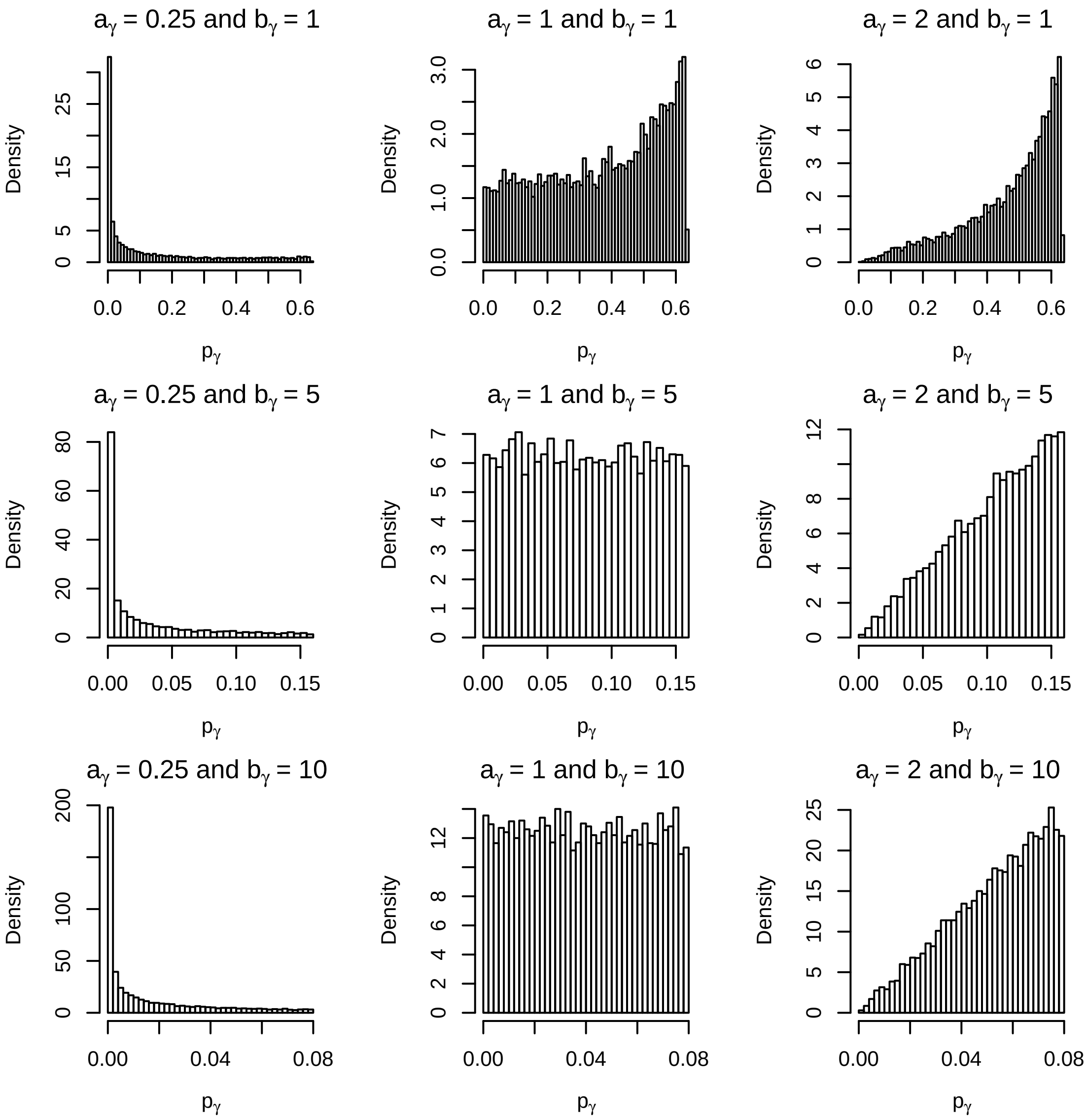


FIGURE 4.1: Distribution of the prior probability of non-shrinkage of the local basis coefficients, $p_\gamma := Pr(\gamma_{i,l} \neq 0 \mid K_\gamma)$, for different choices of Pareto hyperparameters a_γ and b_γ.

probability of any simple periodicity (4, 6, 8 hours, etc.) by counting the proportion of posterior samples for which $\{\theta_{i,2m-1}, \theta_{i,2m}\}$ are *not* shrunk to zero while the remaining $\boldsymbol{\theta}_s$'s are switched off. For example, the posterior probability that protein

i is circadian can be computed as

$$P(\text{Protein } i \text{ is circadian}) = \frac{1}{TS}\sum_{g=1}^{TS} \mathbb{1}(\{\theta_{i,l}^{(g)}\}_{l=1}^{2q-2} \equiv \mathbf{0} \text{ and } \{\theta_{i,2q-1}^{(g)}, \theta_{i,2q}^{(g)}\} \neq \mathbf{0}) \quad (4.20)$$

If we were interested in quantifying the probability of a protein being periodically expressed without making any specific reference to its period, we could simply count the proportion of posterior samples for which any pair $\{\theta_{i,2m-1}, \theta_{i,2m}\}$ is *not* shrunk whereas the remaining parameters are switched off. In symbols,

$$P(\text{Protein } i \text{ is periodic}) = \frac{1}{TS}\sum_{g=1}^{TS} \mathbb{1}\left\{\begin{array}{c} [(\theta_{i,1}^{(g)}, \theta_{i,2}^{(g)}) \neq \mathbf{0} \text{ and } (\theta_{i,l}^{(g)})_{l=3}^{q} \equiv \mathbf{0}] \text{ or} \\ [(\theta_{i,3}^{(g)}, \theta_{i,4}^{(g)}) \neq \mathbf{0} \text{ and } (\theta_{i,l}^{(g)})_{l\in\{1,2,4,\ldots,q\}} \equiv \mathbf{0}] \text{ or} \\ \vdots \\ [(\theta_{i,l}^{(g)})_{l=1}^{2q-2} \equiv \mathbf{0} \text{ and } (\theta_{i,2q-1}^{(g)}, \theta_{i,2q}^{(g)}) \neq \mathbf{0}] \end{array}\right\} \quad (4.21)$$

Biologists are interested in identifying clock proteins without incurring into too many false discoveries. Then, we need to compile a list of proteins for which the hypothesis of 24 hours periodicity is probably true, and we want the list to be as large as possible while bounding the rate of false discoveries by some threshold, say k^*. We can rank the proteins according to increasing values of $\beta_i = 1 - \Pr(\text{Protein } i \text{ is circadian})$ and declare all proteins with β_i below a threshold, κ, as clock-controlled proteins

$$\beta_i^* = \mathbb{1}(\beta_i \leqslant \kappa), \quad (4.22)$$

where β_i^* is an indicator for the decision to report protein i as circadian. Müller et al. (2004) show that (4.22) is the optimal decision rule under several loss functions that combine false negative and false discovery counts and/or rates, and the choice of the loss function determines the specific value of κ. In addition, the authors show that the result is true for any probability model with non-zero prior probability for periodic and non-periodic expression. In particular, the probability model can include dependence across proteins.

Given the data, the expected number of false discoveries is

$$C(\kappa) = \sum_i \beta_i \mathbb{1}[\beta_i \leqslant \kappa]$$

since β_i is the conditional probability that identifying protein i as circadian creates a type I error. Hereafter we follow Newton et al. (2004) and choose a data-dependent $\kappa \leqslant 1$ as large as possible such that $C(\kappa)/|J| \leqslant k^*$, where $|J| > 0$ is the size of the list. So, $C(\kappa)/|J|$ is the expected rate of false discoveries given the data.

4.1.5 Inference on phase and amplitude

In addition to period estimation, phase and amplitude of rhythmic transcripts must be accurately estimated. Grouping rhythmic transcripts by phase may suggest a common underlying regulatory mechanism. Also, the most robust cyclic proteins can be identified by amplitude. By making use of standard results from Fourier analysis, any simply periodic function $A\cos(\frac{2\pi}{w}t-\psi)$ can be expressed in an essentially unique manner as

$$A\cos\left(\frac{2\pi}{w}t-\psi\right) = A_1\sin\left(\frac{2\pi}{w}t\right) + A_2\cos\left(\frac{2\pi}{w}t\right)$$

The function above is said to have amplitude A and phase shift ψ. Therefore, the de-trended and centered true signal for protein i at time t_j can be written as

$$\begin{aligned} f_i(t_j) &= \sum_{m=1}^{q} A_{i,m}\cos\left(\frac{2\pi}{w_m}t_j-\psi_{i,m}\right) && (4.23)\\ &= \sum_{m=1}^{q}\left[\theta_{i,2m-1}\sin\left(\frac{2\pi}{w_m}t_j\right)+\theta_{i,2m}\cos\left(\frac{2\pi}{w_m}t_j\right)\right], && (4.24)\end{aligned}$$

where $A_{i,m}$ and $\psi_{i,m}$ denote the amplitude and phase of the oscillation with period length w_m. We need to identify $A_{i,m}$ and $\psi_{i,m}$. If we write out

$$\cos\left(\frac{2\pi}{w_m}t_j-\psi_{i,m}\right) = \sin\left(\frac{2\pi}{w_m}t_j\right)\sin\psi_{i,m}+\cos\left(\frac{2\pi}{w_m}t_j\right)\cos\psi_{i,m}$$

we see that we must have

$$A_{i,m} \sin \psi_{i,m} = \theta_{i,2m-1} \qquad \text{and} \qquad A_{i,m} \cos \psi_{i,m} = \theta_{i,2m}$$

Therefore, we get

$$A_{i,m} = \sqrt{\theta_{i,2m-1}^2 + \theta_{i,2m}^2}, \quad \text{and} \tag{4.25}$$

$$\psi_{i,m} = \tan^{-1}\left(\frac{\theta_{i,2m-1}}{\theta_{i,2m}}\right). \tag{4.26}$$

The sets of period, amplitude, and phase $\{w_m, A_{i,m}, \psi_{i,m}\}$, $m = 1, \ldots, q$, provide a complete description of the true process $f_i(t_j)$ underlying the observed oscillation.

4.2 Posterior update

Given the observed data $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^p$, we wish to infer the periodic basis functions coefficients $\{\boldsymbol{\theta}_i\}_{i=1}^p$, the local basis functions coefficients $\{\boldsymbol{\gamma}_i\}_{i=1}^p$, the factor loading matrix $\mathbf{\Lambda}$, the $T \times k$ matrix of latent factors $\boldsymbol{\eta}$, and all hyperparameters. We use Gibbs sampling by successively drawing samples from the full conditional distributions of each parameter in turn, given all other parameters.

The conditional distribution of $\mathbf{Y}$ implied by (4.5) is

$$\mathbf{Y}|\mathbf{B}, \mathbf{C}, \mathbf{\Theta}, \mathbf{\Gamma}, \boldsymbol{\eta}, \mathbf{\Lambda}, \mathbf{\Sigma} = \prod_{i=1}^{p} \mathrm{N}(\mathbf{y}_i | \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2 \boldsymbol{I}_T), \tag{4.27}$$

and the likelihood function is

$$
\begin{aligned}
&P(\mathbf{Y}, \boldsymbol{\Theta}, \boldsymbol{\Gamma}, \tilde{\boldsymbol{\Theta}}, \tilde{\boldsymbol{\Gamma}}, \boldsymbol{\Sigma}, \boldsymbol{\eta}, \boldsymbol{\Lambda}, \boldsymbol{\phi}, \boldsymbol{\tau}, \boldsymbol{\varpi}, \boldsymbol{\varpi}^*) = \qquad (4.28)\\
&\prod_{i=1}^{p} \Big\{ \mathrm{N}(\mathbf{y}_i | \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2 \boldsymbol{I}_T) \mathrm{Ga}(\sigma_i^{-2} | a_\sigma, b_\sigma) \times \\
&N_k \left[\boldsymbol{\lambda}_i^\top | \mathbf{0}, \boldsymbol{D}_i(\boldsymbol{\phi}, \boldsymbol{\tau}) \right] p(\boldsymbol{\phi}|\rho) p(\boldsymbol{\tau}|a_1, a_2) \prod_{j=1}^{T} N_k(\boldsymbol{\eta}_j | \mathbf{0}, \boldsymbol{I}_k) \times \\
&N_{2q}(\tilde{\boldsymbol{\theta}}_i | \mathbf{W}\boldsymbol{\lambda}_i, \mathbb{V}\mathrm{ar}(\boldsymbol{\alpha}_i^\theta)) \times N_{\tilde{T}}(\tilde{\boldsymbol{\gamma}}_i | \mathbf{Z}\boldsymbol{\lambda}_i, \mathbb{V}\mathrm{ar}(\boldsymbol{\alpha}_i^\gamma)) \times p(K_\theta) \times p(K_\gamma) \times \\
&\prod_{j=1}^{2q} N_k(\mathbf{W}_j | \mathbf{0}, \boldsymbol{I}_k) \times \prod_{j=1}^{\tilde{T}} N_k(\mathbf{Z}_j | \mathbf{0}, \boldsymbol{I}_k) \times p(\boldsymbol{\varpi}) \times p(\boldsymbol{\varpi}^*) \Big\},
\end{aligned}
$$

where $p(\boldsymbol{\phi}|\rho)$ and $p(\boldsymbol{\tau}|a_1, a_2)$ are the densities of prior distributions induced by MGPSP on vectors of all $\{\phi_{ih}\}_{i=1,\dots,p;\ h=1,\dots,k}$ and all $\{\tau_h\}_{h=1,\dots,k}$, respectively, and $p(\boldsymbol{\varpi})$ and $p(\boldsymbol{\varpi}^*)$ are the densities of prior distributions induced on vectors of all $\{\varpi_{i,m}\}_{i=1,\dots,p;\ m=1,\dots,q}$ and $\{\varpi^*_{i,l}\}_{i=1,\dots,p;\ l=1,\dots,\tilde{T}}$, respectively.

In what follows we use "–" to denote the "rest" of the model, i.e. all random variables not explicitly mentioned in the current state of the Markov Chain. Using the introduced notations we describe a MCMC algorithm for simulation of the full joint posterior distribution of the model parameters.

- *Update of* $\mathbf{W}$*:* We place a conjugate normal prior on the columns of the $k \times 2q$ matrix $\mathbf{W}^\top$, so $\mathbf{W}_l \sim N_k(\mathbf{0}, \boldsymbol{I}), l = 1, \dots, 2q$. This is equivalent to a prior on the rows of matrix $\mathbf{W}$, $\mathbf{W}_l^\top$. Conditioning on the current estimate of $\tilde{\theta}_{i,l} \sim N(\boldsymbol{\lambda}_i^\top \mathbf{W}_l, 1)$ and other model parameters, the posterior update of $\mathbf{W}_l$ is

$$
\mathbf{W}_l \mid - \sim N_k \left(\left(\sum_{i=1}^{p} \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^\top + \boldsymbol{I} \right)^{-1} \left(\sum_{i=1}^{p} \tilde{\theta}_{i,l} \boldsymbol{\lambda}_i \right), \left(\sum_{i=1}^{p} \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^\top + \boldsymbol{I} \right)^{-1} \right)
$$

- *Update of* $\mathbf{Z}$*:* We place a conjugate normal prior on the columns of the $k \times \tilde{T}$ matrix $\mathbf{Z}^\top$, so $\mathbf{Z}_l \sim N_k(\mathbf{0}, \boldsymbol{I}), l = 1, \dots, \tilde{T}$. This is equivalent to a prior on the rows of matrix $\mathbf{Z}$, $\mathbf{Z}_l^\top$. Conditioning on the current estimate of $\tilde{\gamma}_{i,l} \sim N(\boldsymbol{\lambda}_i^\top \mathbf{Z}_l, 1)$ and other model parameters, the posterior update of $\mathbf{Z}_l$ is

$$\mathbf{Z}_l \mid - \sim N_k \left(\left(\sum_{i=1}^{p} \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^\top + \boldsymbol{I} \right)^{-1} \left(\sum_{i=1}^{p} \tilde{\gamma}_{i,l} \boldsymbol{\lambda}_i \right), \left(\sum_{i=1}^{p} \boldsymbol{\lambda}_i \boldsymbol{\lambda}_i^\top + \boldsymbol{I} \right)^{-1} \right)$$

- *Update of* $\boldsymbol{\lambda}_i^\top$*:* We place a MGPSP on row i of $\boldsymbol{\Lambda}$(equivalently, column i of $\boldsymbol{\Lambda}^\top$) as in (4.11). The likelihood contribution factorizes as

$$L(\boldsymbol{\lambda}_i | \tilde{\boldsymbol{\Theta}}, \boldsymbol{\Theta}, \tilde{\boldsymbol{\Gamma}}, \boldsymbol{\Gamma}, \boldsymbol{\eta}, \boldsymbol{\Sigma}, \mathbf{W}, \mathbf{Z}) \propto N_T(\mathbf{y}_i | \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2 \boldsymbol{I}) \times \\ \times N(\tilde{\boldsymbol{\theta}}_i | \mathbf{W}\boldsymbol{\lambda}_i, \mathbb{V}\mathrm{ar}(\boldsymbol{\alpha}_i^\theta)) \times N(\tilde{\boldsymbol{\gamma}}_i | \mathbf{Z}\boldsymbol{\lambda}_i, \mathbb{V}\mathrm{ar}(\boldsymbol{\alpha}_i^\gamma)) \tag{4.29}$$

We assume $\mathbb{V}\mathrm{ar}(\boldsymbol{\alpha}_i^\theta) = \boldsymbol{I}_{2q}$ and $\mathbb{V}\mathrm{ar}(\boldsymbol{\alpha}_i^\gamma) = \boldsymbol{I}_{\tilde{T}}$. The posterior update of $\boldsymbol{\lambda}_i$ is

$$\begin{aligned} \boldsymbol{\lambda}_i \mid - \quad &\sim \quad \mathrm{N}_k \left(\mathbf{V}_{\boldsymbol{\lambda}_i} \mathbf{M}_{\boldsymbol{\lambda}_i}, \mathbf{V}_{\boldsymbol{\lambda}_i} \right), \quad i = 1, \dots, p, \quad \text{where} \\ \mathbf{M}_{\boldsymbol{\lambda}_i} &= \sigma_i^{-2} \boldsymbol{\eta}^\top (\mathbf{y}_i - \mathbf{B}\boldsymbol{\theta}_i - \mathbf{C}\boldsymbol{\gamma}_i) + \mathbf{W}^\top \tilde{\boldsymbol{\theta}}_i + \mathbf{Z}^\top \tilde{\boldsymbol{\gamma}}_i \\ \mathbf{V}_{\boldsymbol{\lambda}_i} &= \left(\tfrac{1}{\sigma_i^2} \boldsymbol{\eta}^\top \boldsymbol{\eta} + \mathbf{W}^\top \mathbf{W} + \mathbf{Z}^\top \mathbf{Z} + \boldsymbol{D}_i^{-1} \right)^{-1} \end{aligned} \tag{4.30}$$

- *Update of* $\tilde{\boldsymbol{\theta}}_i$*:* We sample the conditional posterior $p(\tilde{\boldsymbol{\theta}}_i \mid -)$ sequentially for $i = 1, \dots, p$ using a Metropolis-Hastings (MH) sampler conditional on the other model parameters. The MH proposal originates from a non-thresholded version of the model. Fixing $\mathbb{1}(||\tilde{\boldsymbol{\theta}}_{i,m}|| \geqslant \varpi_{i,m}) \equiv 1$ for $m = 1, \dots, q$, we take the proposal distribution to be $N(\tilde{\boldsymbol{\theta}}_i \mid \boldsymbol{m}_i, \boldsymbol{M}_i)$ with

$$\begin{aligned} \boldsymbol{M}_i &= \left(\sigma_i^{-2} \mathbf{B}^\top \mathbf{B} + \boldsymbol{I}_{2q} \right)^{-1}, \\ \boldsymbol{m}_i &= \boldsymbol{M}_i \times (\sigma_i^{-2} \mathbf{B}^\top \tilde{\mathbf{y}}_i + \mathbf{W}\boldsymbol{\lambda}_i) \end{aligned}$$

with $\tilde{\mathbf{y}}_i = \mathbf{y}_i - \mathbf{C}\boldsymbol{\gamma}_i - \boldsymbol{\eta}\boldsymbol{\lambda}_i$. The candidate is accepted with probability

$$\alpha(\tilde{\boldsymbol{\theta}}_i, \tilde{\boldsymbol{\theta}}_i^*) = \min\left\{1, \frac{N(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i^* + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2 \boldsymbol{I}_T) N(\tilde{\boldsymbol{\theta}}_i^* \mid \mathbf{W}\boldsymbol{\lambda}_i, \boldsymbol{I}) N(\tilde{\boldsymbol{\theta}}_i \mid \boldsymbol{m}_i, \boldsymbol{M}_i)}{N(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2 \boldsymbol{I}_T) N(\tilde{\boldsymbol{\theta}}_i \mid \mathbf{W}\boldsymbol{\lambda}_i, \boldsymbol{I}) N(\tilde{\boldsymbol{\theta}}_i^* \mid \boldsymbol{m}_i, \boldsymbol{M}_i)}\right\}$$

where $\tilde{\boldsymbol{\theta}}_i$ $(\boldsymbol{\theta}_i)$ is the current estimate and $\tilde{\boldsymbol{\theta}}_{i,m}^*$ $(\boldsymbol{\theta}_{i,m}^* = \tilde{\boldsymbol{\theta}}_{i,m}^* \mathbb{1}(||\tilde{\boldsymbol{\theta}}_{i,m}^*|| \geqslant \varpi_{i,m}))$ is the candidate, with $\boldsymbol{\theta}_i^* = \{\boldsymbol{\theta}_{i,m}^*\}_{m=1}^q$.

- *Update of* $\tilde{\boldsymbol{\gamma}}_i$*:* We sample the conditional posterior $p(\tilde{\boldsymbol{\gamma}}_i \mid -)$ sequentially for $i = 1, \ldots, p$ via MH with proposals obtained from a non-thresholded version of the model. Fixing $\mathbb{1}(|\tilde{\gamma}_{i,l}| \geqslant \varpi_{i,l}^*) \equiv 1$ for $l = 1, \ldots, \tilde{T}$, we take the proposal distribution to be $N(\tilde{\boldsymbol{\gamma}}_i \mid \boldsymbol{n}_i, \boldsymbol{N}_i)$ where

  $$\boldsymbol{N}_i = \left(\sigma_i^{-2} \mathbf{C}^\top \mathbf{C} + \boldsymbol{I}_{\tilde{T}}\right)^{-1},$$

  $$\boldsymbol{n}_i = \boldsymbol{N}_i \times (\sigma_i^{-2} \mathbf{C}^\top \tilde{\mathbf{y}}_i + \mathbf{Z}\boldsymbol{\lambda}_i)$$

  with $\tilde{\mathbf{y}}_i = \mathbf{y}_i - \mathbf{B}\boldsymbol{\theta}_i - \boldsymbol{\eta}\boldsymbol{\lambda}_i$. The MH acceptance probability is

  $$\alpha(\tilde{\gamma}_i, \tilde{\gamma}_i^*) = \min\left\{1, \frac{N(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i^* + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2 \boldsymbol{I}_T) N(\tilde{\gamma}_i^* \mid \mathbf{Z}\boldsymbol{\lambda}_i, \boldsymbol{I}) N(\tilde{\gamma}_i \mid \boldsymbol{n}_i, \boldsymbol{N}_i)}{N(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2 \boldsymbol{I}_T) N(\tilde{\gamma}_i \mid \mathbf{Z}\boldsymbol{\lambda}_i, \boldsymbol{I}) N(\tilde{\gamma}_i^* \mid \boldsymbol{n}_i, \boldsymbol{N}_i)}\right\}$$

  where $\tilde{\boldsymbol{\gamma}}_i$ $(\boldsymbol{\gamma}_i)$ is the current estimate and $\tilde{\boldsymbol{\gamma}}_{i,l}^*$ $(\boldsymbol{\gamma}_{i,l}^* = \tilde{\boldsymbol{\gamma}}_{i,l}^* \mathbb{1}(|\tilde{\boldsymbol{\gamma}}_{i,l}^*| \geqslant \varpi_{i,l}^*))$ is the candidate, with $\boldsymbol{\gamma}_i^* = \{\boldsymbol{\gamma}_{i,l}^*\}_{l=1}^{\tilde{T}}$.

- *Update of* $\varpi_{i,m}$*:* The update can be performed via Gibbs sampling conditioning on the current estimate of $\tilde{\boldsymbol{\theta}}_{i,m} = \{\tilde{\theta}_{i,2m-1}, \tilde{\theta}_{i,2m}\}^\top$ and the other model parameters for $i = 1, \ldots, p$ and $m = 1, \ldots, q$. If $||\tilde{\boldsymbol{\theta}}_{i,m}|| > K_\theta$ (the upper bound of the uniform prior on $\varpi_{i,m}$), the posterior update of $\varpi_{i,m}$ is

  $$\varpi_{i,m} \mid - \sim \text{Unif}(0, K_\theta).$$

  Otherwise, sample

  $$\varpi_{i,m} \mid - \sim \begin{cases} \text{Unif}(0, ||\tilde{\boldsymbol{\theta}}_{i,m}||) & \text{with probability} \quad \pi^* \\ \text{Unif}(||\tilde{\boldsymbol{\theta}}_{i,m}||, K_\theta) & \text{with probability} \quad 1 - \pi^*, \end{cases}$$

with

$$\pi^* = \frac{A}{A+D},$$

$$A = N(\mathbf{y}_i \mid \mathbf{B}_{-m}\boldsymbol{\theta}_{i,-m} + \mathbf{B}_m\tilde{\boldsymbol{\theta}}_{i,m} + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2\boldsymbol{I}_T) \times ||\tilde{\boldsymbol{\theta}}_{i,m}||,$$

$$D = N(\mathbf{y}_i \mid \mathbf{B}_{-m}\boldsymbol{\theta}_{i,-m} + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2\boldsymbol{I}_T) \times (K_\theta - ||\tilde{\boldsymbol{\theta}}_{i,m}||),$$

with $N(\mathbf{y}_i \mid \boldsymbol{m}, \boldsymbol{v})$ denotes the Gaussian density function with mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{v}$ evaluated at $\mathbf{y}_i$. Matrix $\mathbf{B}_{-m}$ ($\boldsymbol{\theta}_{i,-m}$) corresponds to the matrix of periodic bases (vector of periodic basis coefficients) with columns (components) $m = \{2m-1, 2m\}$ excluded. Instead, $\mathbf{B}_m$ ($\tilde{\boldsymbol{\theta}}_{i,m}$) denotes the $\{2m-1, 2m\}$-th columns of matrix $\mathbf{B}$ (the $\{2m-1, 2m\}$-th components of $\tilde{\boldsymbol{\theta}}_i$).

- *Update of $\varpi^*_{i,l}$:* The update can be performed via Gibbs sampling conditioning on the current estimate of $\tilde{\gamma}_{i,l}$ and the other model parameters for $i = 1, \ldots, p$ and $l = 1, \ldots, \tilde{T}$. If $|\tilde{\gamma}_{i,l}| > K_\gamma$ (the upper bound of the uniform prior on $\varpi^*_{i,l}$), the posterior update of $\varpi^*_{i,l}$ is

  $$\varpi^*_{i,l} \mid - \sim \text{Unif}(0, K_\gamma).$$

  Otherwise, sample

  $$\varpi^*_{i,l} \mid - \sim \begin{cases} \text{Unif}(0, |\tilde{\gamma}_{i,l}|) & \text{with probability} \quad \pi^* \\ \text{Unif}(|\tilde{\gamma}_{i,l}|, K_\gamma) & \text{with probability} \quad 1-\pi^*, \end{cases}$$

  with

  $$\pi^* = \frac{E}{E+F},$$

  $$E = N(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}_{-l}\boldsymbol{\gamma}_{i,-l} + \mathbf{C}_l\tilde{\gamma}_{i,l} + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2\boldsymbol{I}_T) \times |\tilde{\gamma}_{i,l}|,$$

  $$F = N(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}_{-l}\boldsymbol{\gamma}_{i,-l} + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2\boldsymbol{I}_T) \times (K_\gamma - |\tilde{\gamma}_{i,l}|)$$

  Matrix $\mathbf{C}_{-l}$ ($\boldsymbol{\gamma}_{i,-l}$) corresponds to the matrix of local bases (vector of local basis coefficients) with column (component) l excluded. Instead, $\mathbf{C}_l$ ($\tilde{\gamma}_{i,l}$) denotes the l-th column of matrix $\mathbf{C}$ (the l-th component of $\tilde{\boldsymbol{\gamma}}_i$).

Further,

$$K_\theta \mid - \quad \sim \quad \text{Pareto}\left(a_\theta + pq, \max\{b_\theta, \max_{i,m}\{\varpi_{i,m}\}_{i=1,m=1}^{p,q}\}\right);$$

$$K_\gamma \mid - \quad \sim \quad \text{Pareto}\left(a_\gamma + p\tilde{T}, \max\{b_\gamma, \max_{i,l}\{\varpi_{i,l}^{*}\}_{i=1,l=1}^{p,\tilde{T}}\}\right);$$

$$\sigma_i^{-2} \mid - \quad \sim \quad \text{Ga}\left(a_\sigma + \frac{T}{2}, b_\sigma + \frac{||\mathbf{y}_i - \mathbf{B}\boldsymbol{\theta}_i - \mathbf{C}\boldsymbol{\gamma}_i - \boldsymbol{\eta}\boldsymbol{\lambda}_i||^2}{2}\right), \quad i = 1, \ldots, p;$$

$$\boldsymbol{\eta}_j \mid - \quad \sim \quad N_k\left[\mathbf{V}_{\boldsymbol{\eta}_j}\mathbf{M}_{\boldsymbol{\eta}_j}, \mathbf{V}_{\boldsymbol{\eta}_j}\right], \quad j = 1, \ldots, T, \quad \text{where}$$

$$\mathbf{M}_{\boldsymbol{\eta}_j} = \boldsymbol{\Lambda}^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}^{(j)} - \boldsymbol{\Theta}\boldsymbol{b}_j - \boldsymbol{\Gamma}\boldsymbol{c}_j),$$

$$\mathbf{V}_{\boldsymbol{\eta}_j} = (\boldsymbol{I}_k + \boldsymbol{\Lambda}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda})^{-1};$$

$$\phi_{ih} \mid - \quad \sim \quad \text{Ga}\left(\frac{\rho + 1}{2}, \frac{\rho + \tau_h \lambda_{ih}^2}{2}\right), \quad i = 1, \ldots, p \quad \text{and} \quad h = 1, \ldots, k;$$

$$\zeta_1 \mid - \quad \sim \quad \text{Ga}\left(a_1 + \frac{pk}{2}, 1 + \frac{1}{2}\sum_{l=h}^{k} \tau_l^{(1)} \sum_{i=1}^{p} \phi_{il}\lambda_{il}^2\right);$$

$$\zeta_h \mid - \quad \sim \quad \text{Ga}\left(a_2 + \frac{p}{2}(k - h + 1), 1 + \frac{1}{2}\sum_{l=1}^{k} \tau_l^{(h)} \sum_{i=1}^{p} \phi_{il}\lambda_{il}^2\right),$$

$$\text{for} \quad h \geqslant 2, \quad \text{where} \quad \tau_l^{(h)} = \prod_{t=1, t\neq h}^{l} \zeta_t \quad \text{for} \quad h = 1, \ldots, k.$$

The three main bottlenecks are in the posterior update of $\{\boldsymbol{\lambda}_i, \tilde{\boldsymbol{\gamma}}_i, \tilde{\boldsymbol{\theta}}_i\}_{i=1}^{p}$, which require looping through the number of variables, p. However, each update i is independent of the others, so loop iterations can be executed in parallel by using function `parfor` in Matlab. Table 4.1 reports the total CPU time (in seconds) per hundred of iterations in Matlab on an Intel(R) Core(TM) i7-2600 machine. The simulation refers to a choice of $q = 5$ and $\tilde{T} = 20$. Larger experiments than those in Table 4.1 face serious time and memory constraints.

Preliminary sensitivity analyses will be required to adjust the priors and other

Table 4.1: CPU time (in seconds) per hundred of iterations required to run the MCMC algorithm of Section 4.2 in Matlab on an Intel(R) Core(TM) i7-2600 machine.

	CPU time	
p	`parfor`	Regular `for` loop
1000	66.80	160.84
5000	1.17e+03	1.75e+03
10000	4.71e+03	5.74e+03
15000	1.06e+04	1.88e+04
20000	1.89e+04	2.40e+04
25000	3.13e+04	3.51e+04

model parameters to provide the best fit to the data. To save on computing time, it might be preferable to run the preliminary analyses on a randomly chosen subset of probes and proceed to the analysis of the complete data set when one is satisfied with the choice of the hyperparameters and other parameter values.

4.3 Simulation studies

4.3.1 Dependence across measurements

We synthesized data from the model, and then used the above framework to infer the model parameters. We simulated $\mathbf{y}_i$, $i = 1, \ldots, p = 500$, from a $T = 24$-dimensional normal distribution with mean $\mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i$ and covariance matrix $\sigma_i^2 \times \boldsymbol{I}_T$, with $\sigma_i^2 = 0.5\ \forall i$. The design matrix $\mathbf{B}$ included the Fourier bases as specified in Section 4.1 with possible periods $\{4, 6, 8, 12, 24\}$ hours, thus $q = 5$, whereas $\tilde{T} = 10$ Gaussian kernels with common bandwidth $\psi = 25$ were chosen for the matrix of local bases $\mathbf{C}$. The true number of factors was set equal to $k = 6$, and the number of non-zero elements in each column of $\boldsymbol{\Lambda}$ were chosen linearly between $2 \times (10 \log p)$ and $10 \log p + 1$. In practice, this resulted in a number of non-zero elements between 99 and 124 across the different columns of $\boldsymbol{\Lambda}$. We randomly allocated the location of the zeros in each column and simulated the non-zero elements independently from a normal distribution with mean 0 and variance 9. The latent factors $\boldsymbol{\eta}$ were independently generated by sampling from a standard normal distribution. The $p \times q$ true latent thresholds for $\boldsymbol{\Theta}$ were independently generated from a Unif$(0, 6)$ whereas the $p \times \tilde{T}$ latent thresholds for $\boldsymbol{\Gamma}$ were independently generated from a Unif$(0, 10)$ to induce sparsity on $\boldsymbol{\Gamma}$ and jitter the curves with only a few, time-localized deviations. The rows of $\mathbf{W}$ ($\mathbf{Z}$) were independently generated by sampling from a standard normal distribution, and the true values of the latent coefficients $\{\tilde{\boldsymbol{\theta}}_i, \tilde{\boldsymbol{\gamma}}_i\}_{i=1}^{p}$ were generated by sampling from their prior distribution given the true values of $\mathbf{W}, \mathbf{Z}$, and $\boldsymbol{\Lambda}$.

We run the Gibbs sampler described in Section 4.2 for 50000 iterations with a burn-in of 20000, and collected every 5th sample to thin the chain. The hyperparameters a_σ and b_σ for σ_i^{-2} were 1 and 0.5, respectively, while $\rho = 3$, $a_1 = 2.1$, $a_2 = 3.1$, $a_\theta = a_\gamma = 1$, $\beta_\theta = 5, \beta_\gamma = 10$ and used $k = 5$ as the starting number of factors.

Of the 500 curves, 22.4% exhibit simple periodicity with periods either $\{4, 6, 8, 12, 24\}$ hours and the remaining profiles either load on more than one Fourier basis or are pure noise. Only 25 of the 500 simulated profiles truly exhibit circadian expression. Therefore, the signal-to-noise ratio is quite weak in this dataset. Figure 4.2 shows the estimated trajectories for the 25 circadian variables: the black line represents the true trajectory, the blue line represents the posterior mean estimate of $\mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\eta}\lambda_i$, and the red dashed lines are the 95% pointwise credible intervals of the same quantity.

Figure 4.3 shows a comparison between the true correlation (left panel) and the estimated correlation structure (right panel). To improve visibility, the plot only reports probes that give rise to true pair-wise correlations of or above (below) 0.90 (-0.90). The correlation structure seems overall slightly under-estimated: this is likely the effect of the MGPS prior on $\boldsymbol{\Lambda}$, which tends to favor small (in magnitude) loadings, as opposed to the wide-support distribution ($N(0, 9)$) used to generate the true

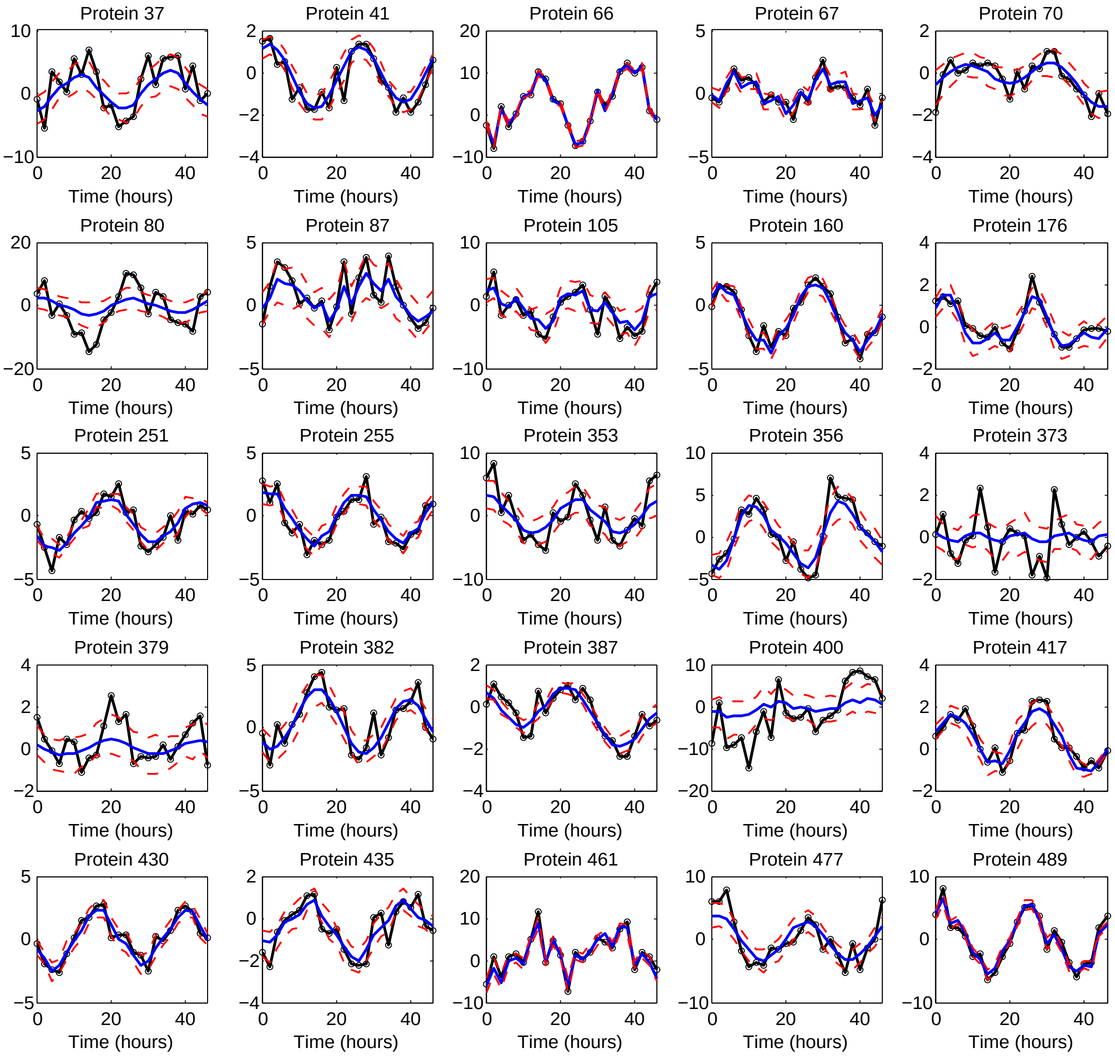


FIGURE 4.2: True (black) and inferred (blue) trajectories of the 25 truly circadian variables in the simulation study of Section 4.3.1, with the horizontal axis corresponding to time. Red lines are the pointwise 95% credible intervals.

non-zero elements on $\mathbf{\Lambda}$. By examining the correlation matrix generated by probes with *estimated* pair-wise correlation of or above (below) 0.80 (-0.80), we notice an almost perfect match with their true correlation structure (Figure 4.4).

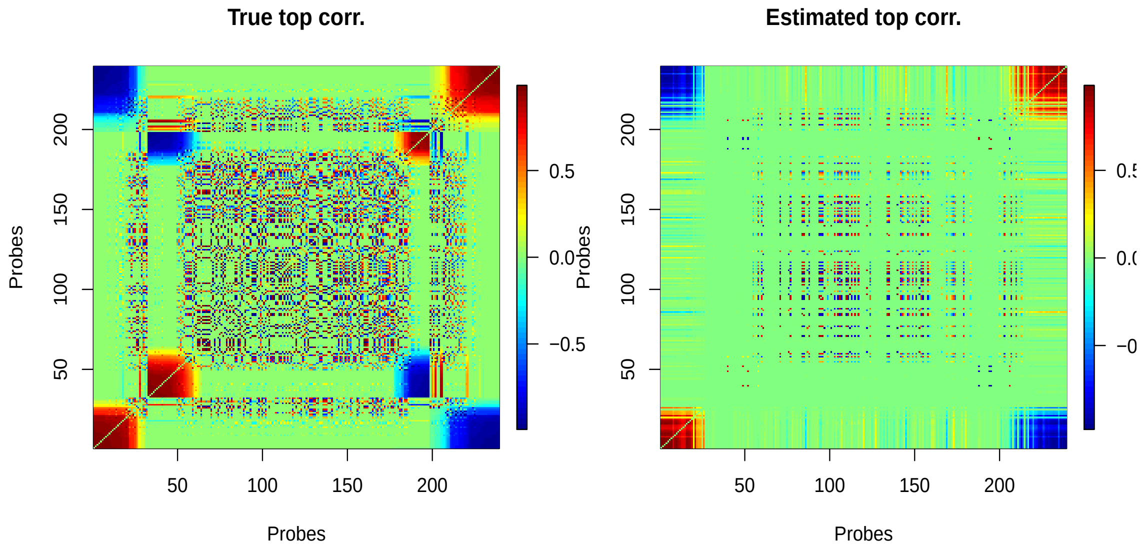


Figure 4.3: Comparison between true and estimated correlation structure among probes with true pair-wise correlation of or above (below) 0.90 (-0.90). To improve visibility, the diagonal of the correlation matrix is set equal to 0.

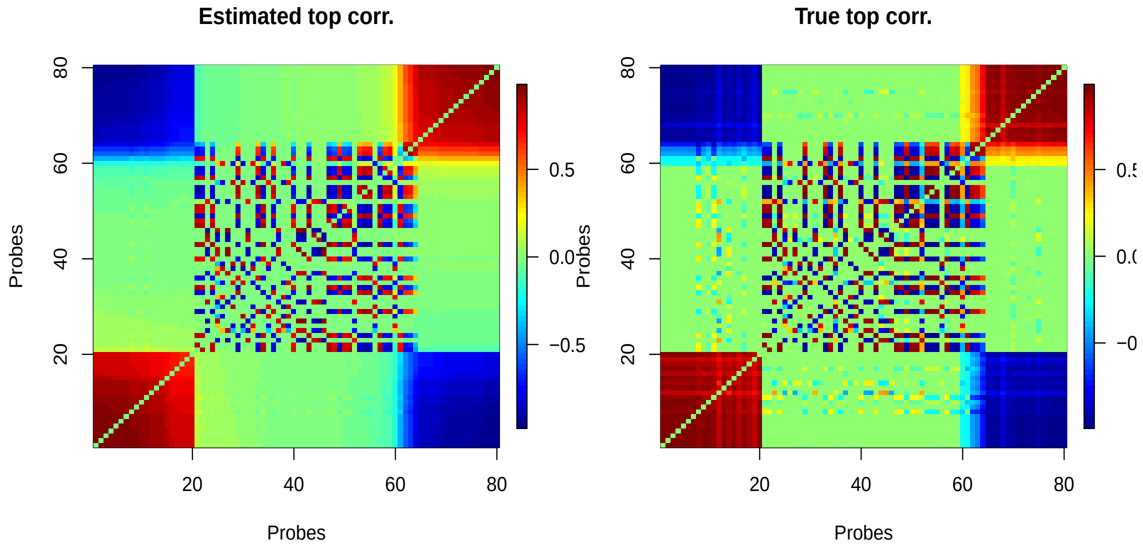


Figure 4.4: Comparison between estimated and true correlation structure among probes with estimated pair-wise correlation of or above (below) 0.80 (-0.80). To improve visibility, the diagonal of the correlation matrix is set equal to 0.

When computing model-parameter summaries, one must address the fact that the state (zero or non-zero) of the periodic and local basis coefficients $\{\boldsymbol{\theta}_i, \boldsymbol{\gamma}_i\}$ may change between collection of samples (it should change between collection of samples if there is good mixing). When aggregating collection samples, reporting the posterior mean for these parameters could be misleading in that the non-zero estimates will bias the overall posterior mean, which is therefore unlikely to be exactly zero even when the true value of the parameter is zero. Table 4.2 reports the proportion of posterior samples for which $\boldsymbol{\theta}_m \equiv \{\theta_{i,2m-1}, \theta_{i,2m}\}, m = 1, \ldots, 4$ are estimated being equal to zero while $\boldsymbol{\theta}_5 \equiv \{\theta_{i,9}, \theta_{i,10}\}$ are estimated being different from zero for the 25 circadian variables. We recall that $\boldsymbol{\theta}_5$ denotes the vector of coefficients of the sine/cosine bases with 24 hours period. The table also shows the estimated circadian probability (Equation 4.20) and quantiles of the inferred phase and amplitude of the oscillation with period length of 24 hours, $\psi_{i,5}$ and $A_{i,5}$.

To assess the performance of the proposed method, we compared our approach with Fisher's g-test (Wichert et al., 2004), robust g-test (Ahdesmäki et al., 2005), and JTK cycle (Hughes et al., 2010). These methods test the hypothesis of "absence of periodicity" (H_0) versus "signal is periodic" (H_1) with unspecified period. Therefore, the comparison is made by evaluating the estimated probability that a protein is periodic (Equation 4.21). We also compared our method to its "independent" version that is,

$$\mathbf{y}_i = \mathbf{B}\boldsymbol{\theta}_i + \mathbf{C}\boldsymbol{\gamma}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \sigma_i^2 \boldsymbol{I}) \tag{4.31}$$

Model (4.31) still accommodates local deviations and can detect periodicity, but does not accommodates dependence across variables by blocking inference on $\boldsymbol{\Lambda}$ which is kept fixed to zero. Therefore, it is the default version of our model in scenarios of independence across variables.

For every method, we ordered the p-values (or the estimated circadian probability for our approach) which show how strong the evidence is against the hypothesis of

Table 4.2: Simulation study as of Section 4.3.1: columns refer to the different periodic basis parameters and rows to the truly circadian variables. Columns 2-5 report the proportion of posterior samples for which $\boldsymbol{\theta}_m = \{\theta_{i,2m-1}, \theta_{i,2m}\}$ are estimated being equal to zero for $m = 1, \dots, 4$, whereas column 6 reports the proportion of posterior samples for which $\boldsymbol{\theta}_5 = \{\theta_{i,9}, \theta_{i,10}\}$ are estimated being different from zero. Vector $\boldsymbol{\theta}_5$ contains the coefficients of the oscillations with period length of 24 hours. Column 7 reports the estimated probability of a protein being circadian, i.e. the proportion of posterior samples with $\{\theta_{i,m}\}_{m=1}^{2q-2} \equiv 0$ and $\{\theta_{i,2q-1}, \theta_{i,2q}\} \neq \mathbf{0}$. The last two columns report the true amplitude and phase versus the [2.5, 50, 97.5]% quantiles of estimated amplitude and phase for the oscillation with period length of 24 hours (brackets). Ranking is by posterior circadian probability.

Gene	$\boldsymbol{\theta}_{i,1}$	$\boldsymbol{\theta}_{i,2}$	$\boldsymbol{\theta}_{i,3}$	$\boldsymbol{\theta}_{i,4}$	$\boldsymbol{\theta}_{i,5}$	P(Circ)	$A_{i,5}$ (true/est)	$\psi_{i,5}$ (true/est)
66	100	100	100	100	100	1	4.84 [4.1, 4.5, 4.9]	0.48 [0.47, 0.50, 0.52]
251	100	99.7	95.1	99.9	100	0.95	1.46 [1.4, 1.4, 1.4]	1.53 [-1.28, -1.28, -1.28]
489	98.5	98.7	99.3	97.5	100	0.94	4.09 [3.22, 3.73, 4.46]	0.35 [0.25, 0.38, 0.45]
387	96	97.3	100	100	99.9	0.93	0.75 [0.86, 0.86, 0.98]	0.14 [-0.67, -0.67, -0.56]
41	91.4	99	99.6	97.6	100	0.88	1.26 [1.09, 1.37, 1.66]	0.54 [0.34, 0.49, 0.76]
417	87.4	99	100	100	100	0.87	1.44 [1.23, 1.23, 1.23]	1.23 [1.1, 1.1, 1.1]
382	95.8	93.6	94.1	94.4	100	0.80	1.38 [1.68, 2.29, 2.78]	0.87 [0.58, 0.86, 1.1]
477	92.8	94.1	94.2	87.7	100	0.72	4.09 [2.19, 3.14, 4.26]	0.33 [0.01, 0.41, 0.65]
255	75.8	92.6	98.8	98.9	100	0.69	2.43 [1.73, 1.89, 2.4]	0.26 [0.18, 0.44, 0.6]
160	66.5	95.7	99.1	93.4	100	0.59	1.85 [1.43, 1.7, 2.13]	1.07 [0.65, 0.9, 1.12]
70	94.9	98.1	99	98.3	81.4	0.57	0.92 [0.45, 0.7, 1.12]	-0.39 [-1.32, -0.83, 1.51]
435	97.5	99.4	62.4	96.3	99.8	0.57	1.48 [0.71, 0.73, 1.35]	0.47 [-0.01, 0.35, 0.69]
87	92.2	75.6	83.4	97.5	93.9	0.53	1.66 [0.77, 1.28, 1.94]	1.3 [-1.56, -1.04, 1.54]
37	87.9	85	79	88.9	98.8	0.53	4.96 [1.44, 2.78, 4.11]	-0.33 [-0.67, -0.17, 0.32]
430	93.4	99	56.5	98.1	100	0.52	1.84 [1.91, 2.09, 2.3]	1.09 [0.98, 1.16, 1.27]
353	88	72.6	88.8	90.8	98.8	0.51	4.11 [1.71, 2.76, 4.02]	-0.08 [-0.42, 0.11, 0.56]
80	82.3	81.9	81.6	72.5	87.6	0.35	5.73 [1.11, 2.75, 4.42]	0.15 [-0.42, 0.48, 1.21]
461	99.6	33.4	99.1	98.4	100	0.32	3.8 [2.86, 4.16, 4.16]	0.21 [-0.08, 0.03, 0.17]
379	93.8	99	98.4	91.9	36.8	0.28	0.56 [0.55, 1.03, 1.3]	-0.75 [-1.42, -1.4, -0.36]
400	79.7	74.4	82.6	82.3	64.4	0.26	4.06 [0.78, 2.21, 4.54]	-1.41 [-1.54, 0.94, 1.54]
356	93.8	95.7	93.4	26.1	100	0.20	3.79 [2.26, 3.45, 4.03]	-0.05 [-0.37, 0.04, 0.17]
176	95.3	98.7	98.2	6.7	100	0.06	1.18 [0.67, 0.92, 1.35]	0.67 [0.14, 0.56, 1.02]
105	75.8	19	96.5	37.7	99.7	0.06	2.56 [1.15, 1.71, 3.62]	-0.08 [0.12, 0.7, 0.97]
67	92.1	5.5	99.9	67.7	100	0.04	1.27 [0.85, 1.2, 1.64]	-1.31 [-1.5, -1.17, -0.86]
373	86.8	96.6	98.2	69.7	3	0.02	0.49 [0.29, 0.56, 1.17]	-0.28 [-0.9, 0.12, 1.37]

absence of periodic signal. Based on this ordering, we picked-up the first N_i variables from the ordered lists and compared the proportion the true positives, namely the proportion of periodic variables correctly identified as periodic, and the proportion of true negatives, namely the proportion of non-periodic variables correctly classified as non-periodic. Since predictions were periodic or non-periodic, a well-suited binary classification, we applied receiver operating characteristic (ROC) curves to compare the performances of the four algorithms by varying N_i sequentially from $i = 1$ to $i = p$. The performance is measured by the area under the ROC curve criterion, with the larger area the better method (Figure 4.5). It is evident that our approach outperforms methods that do not directly accommodate dependence across variables. The grey ROC curve refers to a second chain for our model initialized at over-dispersed starting values; it is used to check the reproducibility of the results. The right panel of Figure 4.5 shows the progression of the false discovery rate (FDR) as function of the true positive rate (power). The FDR is defined as the ratio between false positives, namely the number of variables falsely declared as periodic, and the number of positives, namely the total number of variables declared periodic. The spike at 0 for Fisher's g-test and robust g-test means that the protein with smallest p-value, thus the first selected as periodic, is in fact a false positive. The more variables we include in the list of periodic variables, the more the power increases. A value of power equal to 1 corresponds to detection of all truly periodic variables as periodic. For large values of power, our model achieves lower FDR than methods which do not accommodate dependence.

Several chains were run to assess the sensitivity of the results to different choices of $a_\theta, a_\gamma, b_\theta, b_\gamma$, and other model parameters. In all cases, our approach achieved better performance than methods not accommodating dependence across variables.

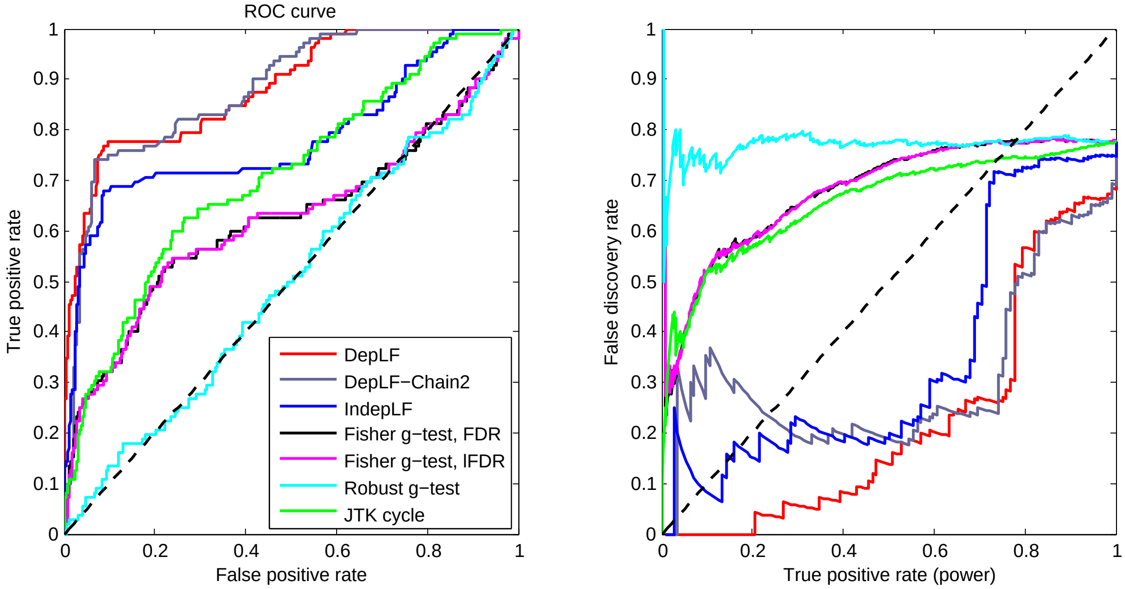


FIGURE 4.5: Left panel: ROC curve for identifying periodic signals in the simulated example with dependence across variables. Right panel: progression of false discovery rate (FDR) for different levels of true positive rate. Two chains initialized at over-dispersed starting values were run to assess the reproducibility of the results, and these chains correspond to the red and grey lines. The blue line corresponds to the "independent" version of our method as in (4.31). Multiple testing for Fisher's g-test is done using tail area-based FDR and density-based local FDR (lFDR).

4.3.2 Independence across measurements

For any modeling approach which accommodates dependence across variables, one concern is that if the true profiles are indeed independent, whether the "unnecessarily sophisticated" dependent-modeling approach can perform as good as the "correct, independent" model.

To test the performance of our proposed method in such case, we simulated sample paths from the model but fixed $\mathbf{\Lambda} \equiv \mathbf{0}$ such that at any time point $j = 1, \ldots, T$ the variables were independent of each other, $\mathbf{y}^{(j)} \sim N(\mathbf{\Theta b}_j + \mathbf{\Gamma} \boldsymbol{c}_j, \mathbf{\Sigma})$, with $\mathbf{\Sigma}$ diagonal. We increased σ_i^2 to $1, \forall i$ and generated the $p \times q$ true latent thresholds for $\mathbf{\Theta}$ independently from a Unif$(0, 5)$. All remaining parameters were generated as described in Section 4.3.1. Of the $p = 500$ simulated trajectories, 4% were circadian and 78 were periodic with possible periods either $4, 6, 8, 12$ or 24 hours.

We run the Gibbs sampler described in Section 4.2 for 50000 iterations with a burn-in of 20000, and collected every 5th sample to thin the chain. The hyperparameters a_σ and b_σ for σ_i^{-2} were 1 and 0.5, respectively, while $\rho = 3$, $a_1 = 2.1$, $a_2 = 3.1$, $a_\theta = a_\gamma = 1, \beta_\theta = \beta_\gamma = 5$ and used $k = 4$ as the starting number of factors.

The additional complexity does not affect our method, which still outperforms Fisher's g-test, robust g-test and JTK cycle and performs at least as well as its corresponding independent version (Figure 4.6). The good performance has to be attributed to the shrinkage property of the MGPSP. Figure 4.7 shows side-by-side boxplots of the posterior mean estimate of the factor loadings that is, the posterior means of $\{\mathbf{\Lambda}_{1:p,1}, \mathbf{\Lambda}_{1:p,2}, \ldots, \mathbf{\Lambda}_{1:p,k=6}\}$, where $k = 6$ is the posterior mean of the estimated number of factors. Although this prior can not return exactly zero estimates for the components of $\mathbf{\Lambda}$, the estimated factor loadings are small in magnitude, thus shrinking toward the truth (zero) the contribution of $\boldsymbol{\eta}\boldsymbol{\lambda}_i$ in Equation 4.5.

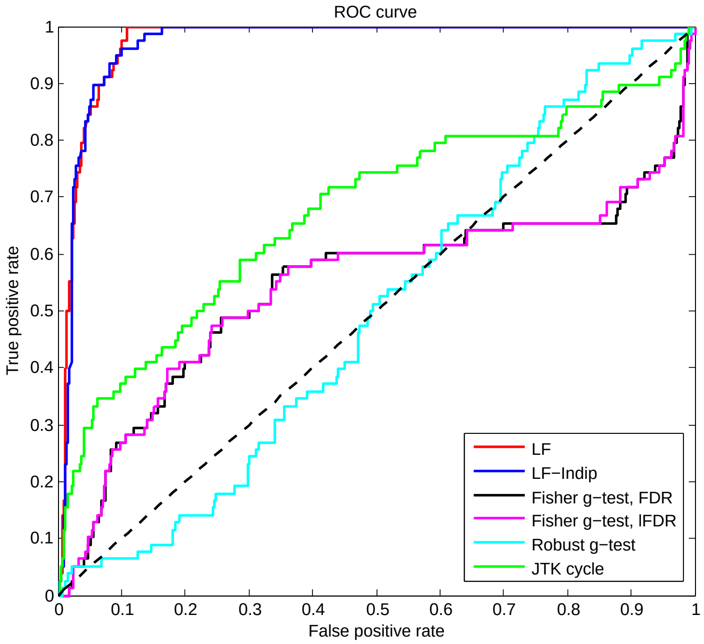


FIGURE 4.6: ROC curve for identifying periodic signals in the simulated example with independence across variables. The blue curve refers to the "independent" version of our method obtained by keeping $\boldsymbol{\Lambda}$ fixed to zero. Multiple testing for Fisher's g-test is done using tail area-based false discovery rate (FDR) and density-based local false discovery rate (lFDR).

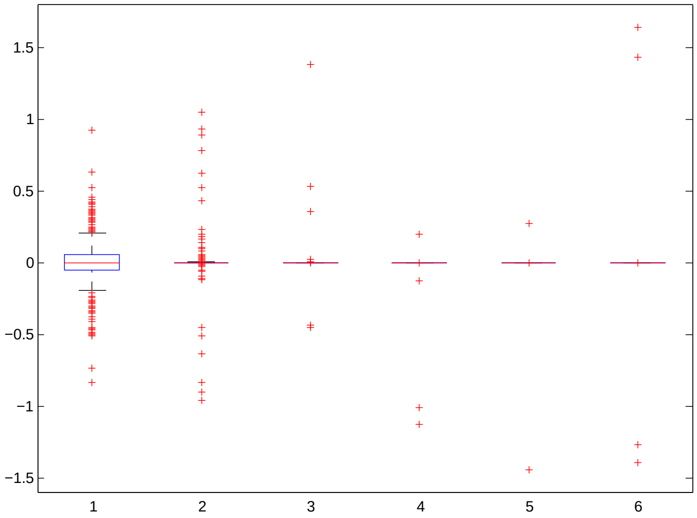


FIGURE 4.7: Side-by-side boxplots of posterior mean estimates of the columns of the factor loading matrix $\boldsymbol{\Lambda}_{:,1}$ (1), . . . , $\boldsymbol{\Lambda}_{:,6}$ (6), where $k = 6$ is the posterior mean estimate of the number of factors. The more these parameters are shrunk toward zero, the closer the model becomes to its "independent" version with $\boldsymbol{\Lambda} \equiv \mathbf{0}$.

4.4 Analysis of *Arabidopsis* circadian expression data

We apply our method to a real dataset generated from the work of Edwards et al. (2006) in the study of the *Arabidopsis* circadian system. The study was designed to detect genes whose expression levels may be connected with the circadian clock. Eight-day-old Columbia seedlings grown under 12-hours-light / 12-hours-dark cycles were transferred to constant light at 22°. Plant samples were harvested at 13 time points, covering two circadian cycles in 4 hours intervals, starting 26 hours after the last dark-light transition. RNA prepared from these samples was analyzed using Affymetrix ATH1 microarrays. In the original study of Edwards et al. (2006), the authors used COSOPT (Straume, 2004) to identify cyclic genes. Of 22810 genes, 3504 genes were considered rhythmic at a significance threshold of pMMC-$\beta < 0.05$ (pMMC-β measures the probability for multiple testing, similarly to the FDR q-value).

The raw expression levels were standardized following standard practice. We run the Gibbs sampler described in Section 4.2 for 40000 iterations with a burn-in of 20000, and collected every 10th sample to thin the chain. The hyperparameters a_σ and b_σ for σ_i^{-2} were 1 and 0.5, respectively, while $\rho = 3$, $a_1 = 2.1$, $a_2 = 3.1$, $a_\theta = a_\gamma = 1, \beta_\theta = 6, \beta_\gamma = 8$ and used $k = 8$ as the starting number of factors. Ten B-splines bases ($\tilde{T} = 10$) were chosen for the matrix of local basis functions. As for the periodic bases, we chose sine/cosine waves with possible periods of $\{8, 12, 16, 20, 24\}$ hours.

Our dependent latent factor approach assigned an estimated circadian probability of or above 0.80 to 1419 genes, thus was more conservative than COSOPT. Of these genes, 1319 were shared with COSOPT. If we want to be less conservative and define circadian all those genes with estimated posterior circadian probability of or above $0.70, 0.60$ or 0.50, respectively, then the number of detected genes increases

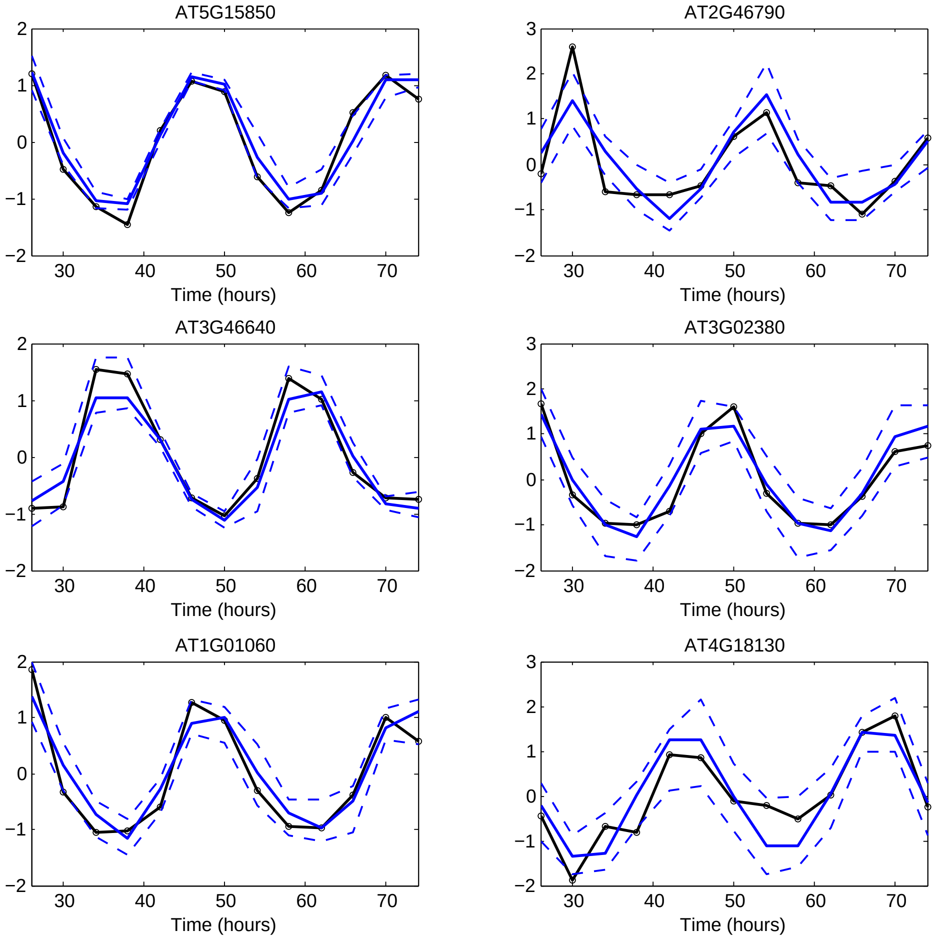


FIGURE 4.8: 6 well-known clock genes by biological knowledge of the *Arabidopsis* dataset that also rank top by posterior probability of being circadian from Equation (4.20). The black trajectory connects the true expression levels, the blue line represents the posterior mean, and dashed blue lines are the pointwise 95% credible intervals. The y-axis denotes the normalized expression levels.

to 1933, 2416, and 2845, respectively. Figure 4.8 shows 6 well-known clock genes by biological knowledge of the *Arabidopsis* dataset that also rank top by posterior probability of being circadian. An inspection of the top circadian genes confirmed sinusoidal patterns through time with a period of approximately 24 hours.

Unfortunately, there is no way to compare the potentially incoherent findings of different statistical approaches on a real dataset in that there is no such a thing as the truth in absence of exact biological validation that supports or contradicts any conclusion. However, a more recent study (Dodd et al., 2007) reported 26 well-known clock-associated genes in *Arabidopsis*. Among these genes are CCA1 (Circadian Clock Associated 1) and LHY (Late Elongated Hypocotyl), which function synergistically in regulating circadian rhythms of *Arabidopsis*, TOC1 (Timing of Cab Expression 1), which contributes to the plant fitness (carbon fixation, biomass) by influencing the circadian clock period, and ELF4 (Early Flowering 4), which accounts for sustained rhythms in the absence of daily light/dark cycles. ELF4 is necessary for light-induced expression of both CCA1 and LHY, and conversely, CCA1 and LHY act negatively on light-induced ELF4 expression. We used these genes as benchmark to evaluate our approach in terms of false negatives for analyzing circadian expression data. We ranked the 22810 genes of the entire *Arabidopsis* genome in order of significance against the null hypothesis of absence of periodicity. The rankings of the 26 known clock genes for the different algorithm is reported in Table 4.3. All the algorithms were able to identify most of the known clock genes from among their top 25% ranked candidates. The complete list of estimated posterior circadian probability for these 26 known clock genes is reported in Table 4.4. Interestingly, the genes with lowest circadian probability (below 0.10) also rank as least likely to be circadian among the 26 known clock-genes with other approaches.

Table 4.3: Summary of rankings of 26 known clock genes in the entire genome of *Arabidopsis*. The ranking of genes was done by posterior circadian probability for our dependent latent factor approach; by pMMC-β for COSOPT; p-value for JTK cycle; by FDR q-value for Fisher's g-test.

Method	Top 1%	Top 5%	Top 10%	Top 25%	Top 60%
Dep. LF	1	10	15	20	25
COSOPT	1	11	16	20	24
JTK cycle	4	11	15	20	25
Fisher's g	2	9	14	20	25

Table 4.4: Posterior estimate of circadian probability for the 26 known clock genes in the entire genome of *Arabidopsis*. Clock genes are identified by their AGI code and Alias name.

AGI	Alias	P(circadian)	AGI	Alias	P(circadian)
At5g15850	COL1	0.99	At5g24470	PRR5	0.75
At2g46790	PRR9	0.96	At1g09570	PHYA	0.69
At3g46640	LUX	0.96	At4g08920	CRY1	0.46
At3g02380	COL2	0.94	At5g02810	PRR7	0.34
At1g01060	LYH	0.92	At4g39260	GRP8	0.23
At4g18130	PHYE	0.91	At2g25930	ELF3	0.20
At2g21660	GRP7	0.88	At1g22770	GI	0.20
At1g04400	CRY2	0.86	At5g59560	SRR1	0.11
At2g40080	ELF4	0.86	At2g18915	LKP2	0.08
At5g61380	TOC1	0.85	At5g35840	PHYC	0.05
At2g46830	CCA1	0.85	At2g46340	SPA1	0.04
At5g60100	PRR3	0.84	At4g16250	PHYD	0.03
At2g18790	PHYB	0.76	At5g57360	ZTL	0.01

4.5 Analysis of mouse liver mRNA data

We apply our method to a real dataset generated from the work of Jouffe et al. (2013). The goal of the study was to assess whether the circadian clock could coordinate the transcription of messenger RNA (mRNA) in mouse liver. In the experiment, C57B1/6J male mice between 10 and 12 weeks of age were used. Mice were maintained under standard animal housing conditions, with free access to food and water and in 12 hours light/12 hours dark cycles. However, mice were fed only at night during 4 days before the experiment to reduce the effects of feeding on rhythm. In the case of rodents, it is in fact during the night period that animals are active and consume food. Liver polysomal and total RNAs were extracted independently from two mice sacrificed every 2 hours during 48 hours. 3 μg of polysomal and total RNAs from each animal from each time point were pooled. The 6 μg of polysomal and total mRNAs were used for the synthesis of biotinylated complimentary RNAs (cRNAs) according to Affymetrix protocol, and the fluorescence signal was analyzed with Affimetrix software (refer to Jouffe et al. (2013) for more details on the study). Data are deposited on the Gene Expression Omnibus database under the reference GSE33726. In the original study, the rhythmic characteristics of the expression of each gene or protein were assessed by a Cosinor analysis (Nelson et al., 1979), and a rhythm was detected if the null hypothesis was rejected with p-value < 0.05. A period of 24 hours was considered a priori. Based on the Cosinor analysis, the authors concluded that the temporal translation of a subset of mRNAs mainly involved in ribosome biogenesis showed evidence of circadian rhythmicity. In addition, the circadian clock appeared to regulate the transcription of ribosomal protein mRNAs and ribosomal RNAs.

The dataset in Jouffe et al. (2013) comprises of 45501 probes among genes and protein products. However, we have no biological validation of some of these probes being circadian at present. Therefore, for illustrative purposes we report the results of our analysis run on a randomly selected subset of $p = 1000$ proteins from the full dataset. The raw expression levels were log-transformed and normalized to zero-mean following standard practice. We run the Gibbs sampler described in Section 4.2 for 50000 iterations with a burn-in of 20000, and collected every 5th sample to thin the chain. Several chains were run to assess the reproducibility of the results under different choices of the parameters values, but no substantial differences were found in the conclusions. Here we report the results of a chain where hyperparameters a_σ and b_σ for σ_i^{-2} were 1 and 0.5, respectively, while $\tilde{T} = 20$ Gaussian kernels with common bandwidth were chosen for the local bases, $\rho = 3$, $a_1 = 2.1$, $a_2 = 3.1$, $a_\theta = a_\gamma = 1, \beta_\theta = 6, \beta_\gamma = 10$ and used $k = 8$ as the starting number of factors. As for the periodic bases, we chose sine/cosine waves with possible periods of $\{4, 6, 8, 12, 24\}$ hours.

Using our model, we ranked the probes by their posterior probability of being periodic (Equation 4.21), and Figure 4.9 shows the top 12 probes. Of the top 20 (50) probes, 19 (42) of them also rank among the top 20 (50) by posterior probability of being circadian. We also tested the probes with Fisher's g-test, JTK cycle, and the independent version of our approach and compared the top 100 probes by evidence against the null hypothesis of absence of periodicity (by p-value or posterior probability of being periodic). Of these top 100 probes, 50 are shared with JTK cycle, 47 with Fisher's g-test, and 61 with the independent version of our method. All together, the four methods agreed on a common set of 36 probes as most likely to be periodic.

A key attractive feature of our model is the accommodation of dependence across variables, thus it becomes of interest to examine the inferred correlation structure from the estimated covariance $\mathbf{\Omega} = \mathbf{\Lambda}\mathbf{\Lambda}^{\top} + \mathbf{\Sigma}$. Most of the estimated correlations are small in magnitude and only 104 pairs of probes have correlation equal or above 0.30 (in absolute value). These "major" correlations are controlled by only 27 (dominant) probes linked (positively or negatively) to each other as shown in Figure 4.10. By inspecting the plot, one can envision two groups of probes: one group with probes $1-13$ in the bottom left corner and the second group with probes $14-27$ in the top right corner. Correlation is positive within each group and negative across groups. To be more conservative, we can further restrict to a set of 8 probes with estimated correlation of or above 0.45 (in absolute value). Again, these 8 probes divide into 2 groups: group 1 with probes 1423069 AT, 1424251 A AT, 1424962 AT, 1426644 AT, 1427200 AT; and group 2 with probes 1419450 AT, 1435068 AT, 1436064 X AT. Probes within each group are positively correlated with each other, whereas the correlation is negative across groups. By examination of the normalized expression levels (Figure 4.11), it becomes evident that probes in each group have similar trajectories. Probes in group 1 exhibit a dip in their normalized expression levels between 10 and 30 hours, as opposed to the peak that probes in group 2 exhibit. Although

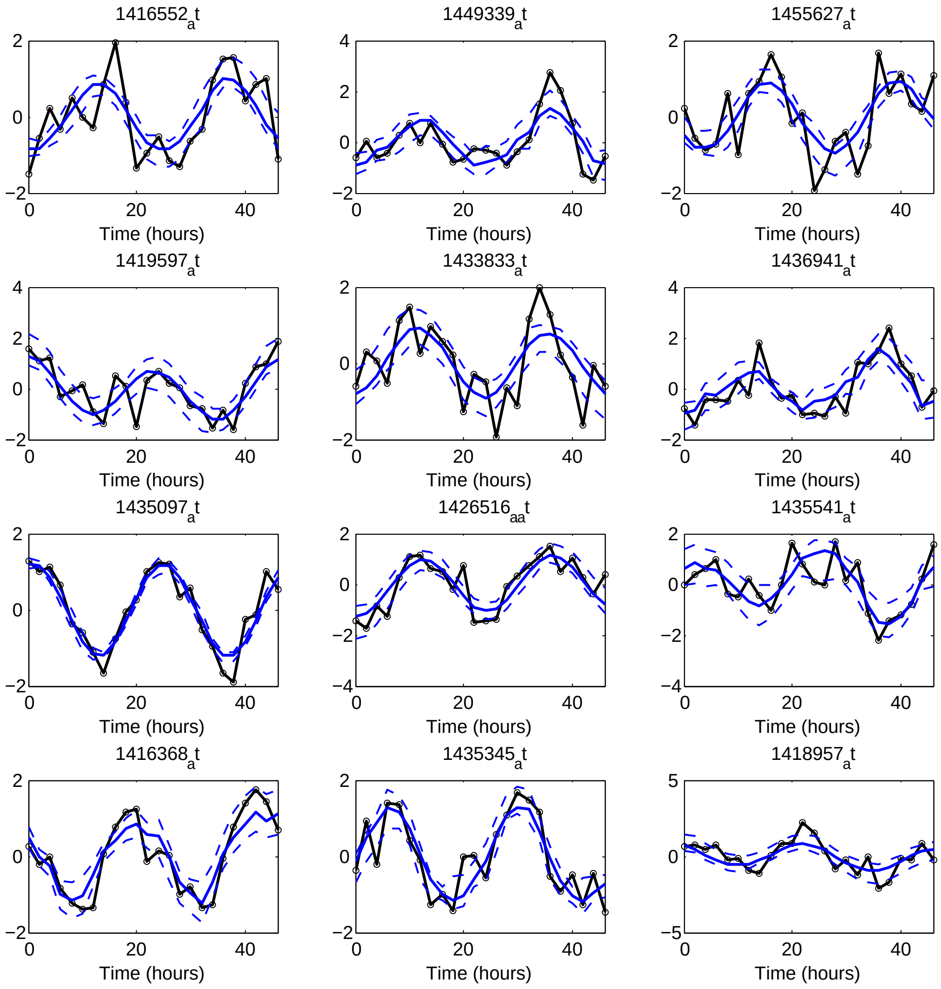


FIGURE 4.9: Top 12 probes in the mouse liver mRNA dataset according to the posterior probability of being periodic from Equation (4.21). The black trajectory connects the true expression levels, the blue line represents the posterior mean, and the dashed lines are the pointwise 95% credible intervals. The y-axis denotes the log-transformed and normalized expression levels.

local deviation seem to emerge in the trajectories of probes in group 2, the sequence of dips and peaks over time seems overall reversed in the two groups.

Table 4.5 shows the estimated amplitude and phase of the oscillation with period length of 24 hours ($A_{i,5}$ and $\psi_{i,5}$) for probes with posterior probability of being circadian of or above 0.80. Probes having the same period and similar phase are expected to have similar trajectories across time and the higher the amplitude, the

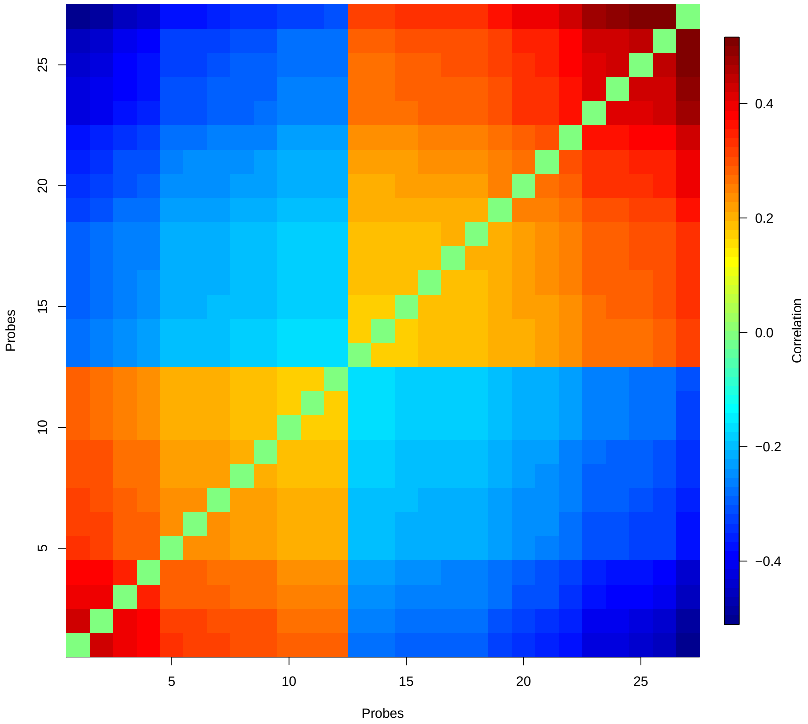


FIGURE 4.10: Correlation structure of the 27 proteins with strongest inferred correlation (of or above (below) 0.30 (-0.30)) in the mouse liver mRNA experiment. To improve visibility, the diagonal of the correlation matrix was set to 0.

more the expression level vibrates/oscillates about its equilibrium. For example, probes 1435207 AT and 1451135 AT have roughly same estimated amplitude and phase, whereas probes 1417063 AT and 1417185 AT have same estimated amplitude but opposite phase. An examination of the raw trajectories for these probes seems to confirm our numerical findings.

4.6 Concluding remarks

A flexible Bayesian methodology for periodicity detection has been developed and applied to large-scale circadian gene expression studies. It employs a Fourier basis

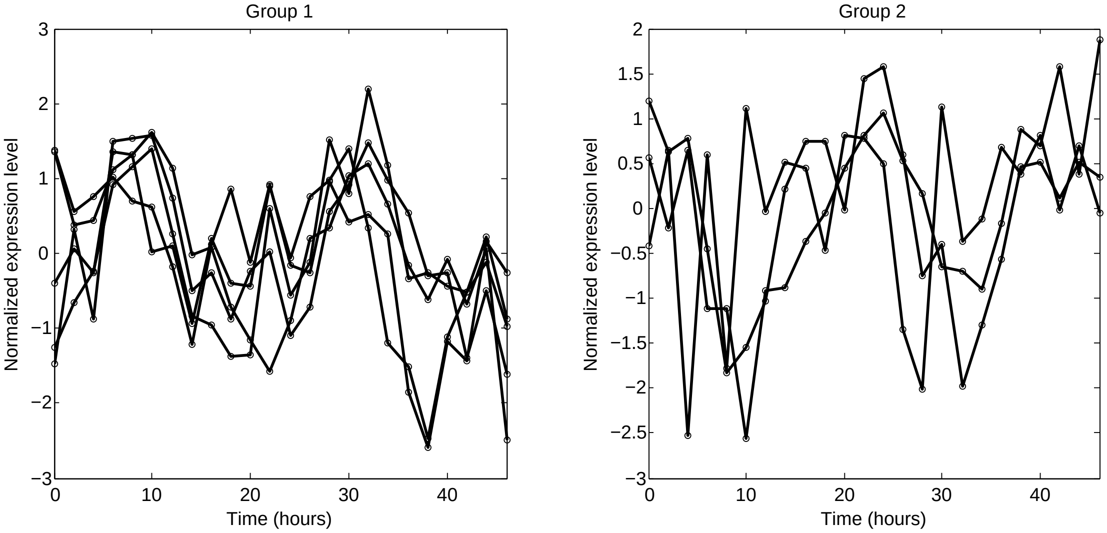


FIGURE 4.11: True trajectories of probes with largest inferred correlations in the mouse liver mRNA experiment. These trajectories were obtained by linearly connecting the observed normalized expression levels. Group 1 includes probes 1423069 AT, 1424251 A AT, 1424962 AT, 1426644 AT, 1427200 AT; group 2 includes probes 1419450 AT, 1435068 AT, 1436064 X AT. Probes within each group are positively correlated with each other, whereas correlation is negative across groups.

expansion with variable selection priors on the basis coefficients to model the time course gene expression trajectories and identify rhythmic genes. The key statistical contribution is to accommodate the potential dependence in the trajectories in terms of latent factors. Our construction allows to infer groups of co-expressed collections of genes and verify relationships within and across groups. Further, accommodating dependence helps identifying weaker patterns appearing is expression profiles by sharing information across genes. Simulation studies show that our construction gives significantly improved performance over widely-used rhythmicity detection techniques that do not directly accommodate for dependence across genes.

The full analysis of the entire mouse liver mRNA dataset (Section 4.5) is challenging within our MCMC procedure given the very large number of simultaneously assessed genes. Among bioinformaticians, it is very common practice to split the dataset into smaller, more manageable subsets. In fact, one could split the data set into multiple pieces (sliding window or clustering as in Anderson et al. (2006)) and then merge results in together. This strategy is not optimal in that it would not take into full account the correlation among all probes, however it would not be worse than treating all probes as independent of each other. Alternatively, one could try to parallelize the procedure presented in this Chapter thought fast GPU computing. This would be an interesting area of future research.

It is important to remark that our findings alone can not be considered definitive proof of whether a gene is clock-regulated. A definitive conclusion can not be made with any statistical approach. A detailed biological investigation of single genes can be an expensive and time consuming process, but it is the only way to validate statistical findings. Our approach should serve as a platform to direct attention toward genes which appear worthwhile of further biological investigation.

Table 4.5: Estimated amplitude and phase of the oscillation with period length of 24 hours for probes with posterior probability of being circadian of or above 0.80 in the mouse liver mRNA study. Probes are ranked from highest to lowest estimate of circadian probability. At every iterations for which the periodic basis coefficients are shrunk to zero with exception for those of the sine/cosine waves of period 24 hours, $\boldsymbol{\theta}_5 = \{\theta_{i,9}, \theta_{i,10}\}$, amplitude and phase are computed as in (4.25)–(4.26). Table shows the median estimate across iterations and the 95% credible interval.

Probes	Amplitude	Phase
1416489 AT	0.92 [0.92, 0.92]	0.19 [0.19, 0.19]
1449325 AT	1.12 [1.12, 1.12]	-0.12 [-0.12, -0.12]
1455587 AT	0.89 [0.89, 0.92]	0.86 [0.86, 0.87]
1419578 AT	1.14 [1.14, 1.14]	-0.17 [-0.17, -0.17]
1433816 AT	0.85 [0.85, 0.85]	-0.02 [-0.02, -0.02]
1436934 S AT	1.25 [1.25,1.25]	-0.32 [-0.32, -0.32]
1435084 AT	1.22 [1.22, 1.22]	0.25 [0.25, 0.25]
1426515 A AT	1.08 [0.84, 1.36]	0.06 [-0.06, 0.32]
1435488 AT	1.55 [0.95, 1.55]	0.45 [-0.14, 0.45]
1416364 AT	0.64 [0.64, 0.64]	-1.23 [-1.23, -1.23]
1435207 AT	1.14 [1.14, 1.14]	1.56 [1.56, 1.56]
1418892 AT	0.75 [0.75, 0.75]	-0.51 [-0.51, -0.51]
1432543 A AT	1.07 [0.93, 1.4]	-0.03 [-0.41, 0.04]
1452687 AT	1.18 [0.72, 1.53]	-0.38 [-1.02, -0.17]
1456170 X AT	1.23 [0.87, 1.54]	-0.09 [-0.44, 0.4]
1423757 X AT	0.87 [0.87, 0.87]	0.03 [0.03, 0.03]
1423202 A AT	0.84 [0.84, 0.84]	-0.25 [-0.25, -0.25]
1441682 S AT	1.25 [1.25, 1.25]	-0.03 [-0.03, -0.03]
1426008 A AT	1.09 [0.66, 1.39]	-0.55 [-1.03, -0.31]
1428845 AT	1 [0.71, 1.47]	1.36 [-1.49, 1.57]
1438629 X AT	1.05 [0.83, 1.23]	-1.39 [-1.51, 1.49]
1448978 AT	0.83 [0.83, 0.86]	-0.33 [-0.44, -0.33]
1445966 AT	1.46 [0.7, 1.46]	1.05 [0.34, 1.05]
1451135 AT	1.29 [0.96, 1.29]	1.52 [1.24, 1.52]
1424962 AT	1.13 [1.13, 1.18]	0.18 [0.01, 0.18]
1417185 AT	1.27 [0.76, 1.65]	-1.38 [-1.53, 1.53]
1424564 AT	1.04 [0.58, 1.58]	0.39 [-0.17, 0.8]
1433555 AT	0.9 [0.79, 1.38]	-0.16 [-0.16, 0.49]
1417063 AT	1.23 [0.83, 1.66]	-1.39 [-1.54, 1.37]
1427661 A AT	1.02 [1.02, 1.02]	-0.83 [-0.83, -0.83]
1437040 AT	1.15 [0.8, 1.15]	0.28 [0.28, 0.33]
1452766 AT	1.09 [0.51, 1.29]	0.3 [-0.03, 1.14]
1434416 A AT	0.75 [0.75, 1.1]	-0.39 [-1, -0.39]
1453271 AT	1.08 [0.71, 1.57]	-0.37 [-0.87, 0.16]
1439260 A AT	1.25 [1.18, 1.25]	0.28 [0.28, 0.87]
1436032 AT	1.48 [0.83, 1.54]	-1.42 [-1.5, 1.5]
1425507 AT	0.79 [0.79, 0.79]	-0.42 [-0.42, -0.42]
1448253 AT	1.16 [0.47, 1.52]	0.63 [0.27, 1.35]
1426631 AT	0.88 [0.63, 1.4]	0.75 [0.29, 1.24]
1455887 AT	1.23 [1.23, 1.23]	-1.23 [-1.23, -1.23]

5

Computer emulation with non-stationary Gaussian processes

Gaussian process models are widely used to emulate propagation uncertainty in computer experiments. Gaussian process emulation sits comfortably within an analytically tractable Bayesian framework. Apart from propagating uncertainty of the input variables, a Gaussian process emulator trained on finitely many runs of the experiment also offers error bars for response surface estimates at unseen input values. This helps select future input values where the experiment should be run to minimize the uncertainty in the response surface estimation. However, traditional Gaussian process emulators use stationary covariance functions, which perform poorly and lead to sub-optimal selection of future input points when the response surface has sharp local features, such as a jump discontinuity or an isolated tall peak. We propose an easily implemented non-stationary Gaussian process emulator, based on two stationary Gaussian processes, one nested into the other, and demonstrate its superior ability in handling local features and selecting future input points from the boundaries of such features.

5.1 Gaussian process emulators

5.1.1 Gaussian process emulation and stationarity

The canonical emulator used for the design and analysis of computer experiments is the Gaussian process. Specifically, for any finite collection of inputs $\boldsymbol{x}^t = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$, $(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_t))^\top$ is jointly distributed as a multivariate normal distribution with mean $\mu(\boldsymbol{x}) = h(\boldsymbol{x})^\top \beta$ and positive definite covariance matrix $C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 K(\boldsymbol{x}, \boldsymbol{x}')$. Although $h(\cdot)$ may be any function on the input space $\mathcal{X}$, hereafter we adopt a linear mean in the inputs, $h(\boldsymbol{x}_i) = [1, x_{i1}, \ldots, x_{ip}]^\top$, with β a vector of unknown parameters. Although the output of a computer model does not generally vary linearly in the inputs, there is often too little prior knowledge on the type of non-linearity.

Clearly, the representation of f as a Gaussian vector makes the computation conceptually straightforward. The conditional distribution of f at a new input $\tilde{\boldsymbol{x}}$, given data $\{\boldsymbol{x}, f(\boldsymbol{x})\}_{1:t} \equiv \{\boldsymbol{X}, \boldsymbol{F}\}$ and model parameters $\boldsymbol{\theta} = \{\beta, \sigma^2, K\}$, is also Gaussian with mean

$$\hat{f}(\tilde{\boldsymbol{x}}) = E[f(\tilde{\boldsymbol{x}}) \mid \{\boldsymbol{x}, f(\boldsymbol{x})\}_{1:t}, \boldsymbol{\theta}] = h(\tilde{\boldsymbol{x}})^\top \beta + k^\top(\tilde{\boldsymbol{x}}) K^{-1}(\boldsymbol{F} - \boldsymbol{X}\beta)$$

and variance

$$\hat{\sigma}^2(\tilde{\boldsymbol{x}}) = \mathrm{var}[f(\tilde{\boldsymbol{x}}) \mid \{\boldsymbol{x}, f(\boldsymbol{x})\}_{1:t}, \boldsymbol{\theta}] = \sigma^2\{K(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}) - k^\top(\tilde{\boldsymbol{x}}) K^{-1} k(\tilde{\boldsymbol{x}})\}$$

where $k^\top(\tilde{\boldsymbol{x}})$ is the $t-$vector whose i-th component is $K(\tilde{\boldsymbol{x}}, \boldsymbol{x}_i), i = 1, \ldots, t$, and K is the $t \times t$ correlation matrix with i, j element $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

The correlation function is crucial in Gaussian process modeling; it is through $K(\boldsymbol{x}, \boldsymbol{x}')$ that we express a belief about how similar $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ should be if $\boldsymbol{x}$ and $\boldsymbol{x}'$ were close in $\mathcal{X}$, thereby we express a belief about the smoothness of f. Although different formulations are possible, in this work we use the separable power correlation function

$$K(\boldsymbol{x}, \boldsymbol{x}') = e^{-\sum_{l=1}^{p} \phi_l (x_l - x_l')^{p_0}}. \tag{5.1}$$

We fix $p_0 = 2$ (product-Gaussian correlation) and infer the correlation range parameters $\{\phi_l\}_{l=1}^{p}$ as part of our estimation procedure. Thus, the correlation is only function of $\boldsymbol{x} - \boldsymbol{x}'$ (stationarity) and a set of roughness (unknown) parameters. van der Vaart and van Zanten (2009) show that the squared-exponential kernel (5.1) can optimally adapt to any smoothness level. Bhattacharya and Dunson (2011a) develop a class of priors for the correlation range parameters which leads to minimax adaptive rates of posterior concentration.

Bayesian inference for Gaussian process emulators proceeds by specifying prior distributions for the model parameters. We use an improper uniform prior on β, $\beta \propto 1$, to reflect weak prior knowledge about these parameters, an inverse-gamma (IG) prior for the scale, $\sigma^2 \sim \mathrm{IG}(a/2, b/2)$, and a log-normal prior for the correlation parameters, $\phi_l \sim \log \mathrm{N}(\mu_\phi, \nu_\phi)$, but other formulations are possible (Gramacy and Lee, 2008). The (marginalized) distribution of f at a new input $\tilde{\boldsymbol{x}}$, conditioned on K and data observed up to time t, is a Student-t distribution with $\hat{\nu} = t - p - 1$ degrees of freedom, mean $\hat{f}(\tilde{\boldsymbol{x}} \mid \{\boldsymbol{x}, f(\boldsymbol{x})\}_{1:t}, K)$, and variance $\hat{\sigma}^2(\tilde{\boldsymbol{x}} \mid \{\boldsymbol{x}, f(\boldsymbol{x})\}_{1:t}, K)$. See Gramacy and Polson (2011) for more details on the derivation of these quantities.

5.1.2 Non-stationary Gaussian process through one latent input

In response to concerns about the adequacy of the stationary assumption, we propose a non-stationary Gaussian process that builds upon the concept of spatial deformation as in Sampson and Guttorp (1992). Specifically, we write

$$Y = f(\boldsymbol{x}, Z), \tag{5.2}$$

thus modeling the simulator as function of both the $p-$dimensional (known) vector of inputs, $\boldsymbol{x} \in \mathcal{R}^p$, and a latent (unknown) input, $Z = g(\boldsymbol{x}) \in \mathcal{R}$, which we infer from the data.

Our formulation relies on two stationary Gaussian processes, one for the function of interest and one for the latent input. Specifically, we assume

$$f \mid \boldsymbol{\theta} \sim \mathrm{GP}(\mu_{\boldsymbol{\theta}}, C_{\boldsymbol{\theta}}), \qquad g \mid \boldsymbol{\theta} \sim \mathrm{GP}(0, \tilde{K}_{\boldsymbol{\theta}}) \tag{5.3}$$

where $\boldsymbol{\theta}$ denotes a vector of model parameters. We model $\mu_{\boldsymbol{\theta}}$ as $\mu_{\boldsymbol{\theta}}(\boldsymbol{x}) = (1, \boldsymbol{x})^{\top}\beta$, and an augmented product-Gaussian correlation form is assumed for $K_{\boldsymbol{\theta}} = \sigma^{-2}C_{\boldsymbol{\theta}}$:

$$K_{\boldsymbol{\theta}}\{\boldsymbol{x}_i, \boldsymbol{x}_j\} = \exp\left\{-\sum_{l=1}^{p} \phi_l (x_{il} - x_{jl})^2 - \phi_{p+1}(Z_i - Z_j)^2\right\}. \tag{5.4}$$

Thus, (5.4) corresponds to the standard (squared-exponential) correlation function of a stationary Gaussian process (5.1) indexed by $p+1$ inputs. Furthermore, we model the correlation function of the Gaussian process on g, i.e. the correlation of the latent level Gaussian process, as

$$\tilde{K}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left\{-\sum_{l=1}^{p} \tilde{\phi}_l (x_{il} - x_{jl})^2\right\}, \tag{5.5}$$

with the scale parameter fixed to 1.

The idea behind the spatial deformation approach is a non-linear transformation of the locations $\boldsymbol{x}^t$ into a latent space within which the correlation structure is stationary (Sampson and Guttorp, 1992). The mapping is done through a Gaussian process prior as in Schmidt and O'Hagan (2000). However, our construction differs from the one in Schmidt and O'Hagan (2000), where $K_{\boldsymbol{\theta}}$ is chosen to correspond to a mixture of Gaussian correlation functions, each of which depends on the Euclidean distance between the latent inputs Z's only. While retaining an elegant formulation, our construction eases a more intuitive interpretation of the problem. We expect the latent process to mimic, at least qualitatively, the behavior of the response. The correlation between points near a sharp, localized feature is weakened since the corresponding distance has been stretched by the latent coordinate.

Clearly, the problem of modeling a non-stationary simulator could be tackled in different ways, e.g. one could proceed to a direct definition a non-stationary covariance function as in Paciorek and Schervish (2004). As opposed to this approach, the latent extension of the input space guarantees positive definiteness of the covariance between observations in the original space and enhances an intuitive interpretation of the problem.

For completeness, we remark that a similar methodological idea was independently developed by Pfingsten et al. (2006) in the context of Gaussian process regression for non-stationary processes.

5.2 Implementation & sequential design

This section presents a sequential Monte Carlo implementation of our non-stationary Gaussian process. The approach relies on particle learning (Lopes et al., 2011), which naturally blends with active learning of the design. The discussion of a two stage approximation of the proposed emulator is deferred to Section 5.6.

First, we need to identify particles $\{S_t^{(i)}\}_{i=1}^N$, which contain all the sufficient information about the uncertainties given data up to time t, with N denoting the total number of particles. The sufficient information necessarily depends upon $[(\boldsymbol{x}_1, f(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_t, f(\boldsymbol{x}_t))]$ [1], thus $\{S_t^{(i)}\}_{i=1}^N = \{(Z_{1:t}, K_t, \tilde{K}_t)^{(i)}\}$, with $Z_{1:t} \equiv (Z_1, \ldots, Z_t)^T$. The correlation functions have been indexed by t to stress their dependency to the data collected up to time t. Particles do not contain β nor σ^2 as these parameters can be marginalized out within our Bayesian construction (Gramacy and Polson, 2011).

Particles are initialized at time $t_0 > p + 1$ with a sample of the unknown parame-

---

[1] For coherence, we remark one should write $f(\boldsymbol{x}_1, Z_1), \ldots, f(\boldsymbol{x}_t, Z_t)$ since the simulator is regarded as function of both the known and latent inputs. In the remainder, however, we will write $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_t)$ to simplify the notation.

ters from their prior distributions. The algorithm for updating particles $\{S_t^{(i)}\}_{i=1}^N$ to $\{S_{t+1}^{(i)}\}_{i=1}^N$ cycles through the following steps:

- *Resample* Generate index $\zeta \sim \text{Multinomial}(w, N)$, with

  $$w^{(i)} = \frac{\pi(f(\boldsymbol{x}_{t+1}) \mid S_t^{(i)})}{\sum_{i=1}^N \pi(f(\boldsymbol{x}_{t+1}) \mid S_t^{(i)})}, \qquad i = 1, \ldots, N,$$

  where $\pi(f(\boldsymbol{x}_{t+1}) \mid S_t^{(i)}) = \pi(f(\boldsymbol{x}_{t+1}) \mid [\boldsymbol{x}, f(\boldsymbol{x})]_{1:t}, K_t^{(i)})$ denotes the probability of observing $f(\boldsymbol{x}_{t+1})$ under a Student-t distribution Gramacy and Polson (2011)

- *Propagate* $S_t^{\zeta(i)}$ to $S_{t+1}^{(i)}$: propagate each particle $S_t^{\zeta(i)}$ to account for $[\boldsymbol{x}_{t+1}, f(\boldsymbol{x}_{t+1})]$
  - The first step requires constructing the "propagated" correlation function of the latent GP, which will be used to sample the latent coordinate at the new input $\boldsymbol{x}_{t+1}$. Thus, we build $\tilde{K}_{t+1}^{(i)}$ from $\tilde{K}_t^{(i)}$ and $\tilde{k}_t^{(i)}(\boldsymbol{x}_{t+1}) = \tilde{K}^{(i)}(\boldsymbol{x}_{t+1}, \boldsymbol{x}_j)$, with $j = 1, \ldots, t$

    $$\tilde{K}_{t+1}^{(i)} = \begin{bmatrix} \tilde{K}_t^{(i)} & \tilde{k}_t^{(i)}(\boldsymbol{x}_{t+1}) \\ \tilde{k}_t^{(i)\top}(\boldsymbol{x}_{t+1}) & \tilde{K}^{(i)}(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1}) \end{bmatrix}$$

  - We obtain $Z_{t+1}^{(i)} = g^{(i)}(\boldsymbol{x}_{t+1})$ from its predictive distribution $g^{(i)}(\boldsymbol{x}_{t+1}) \mid g^{(i)}(\boldsymbol{x}_{1:t}), \tilde{K}_t^{(i)} \sim \mathrm{N}(\mu^{*(i)}, \tilde{K}^{*(i)})$, where the mean and covariance are obtained via standard kriging equations
  - We construct the "propagated" correlation function of f. We build $K_{t+1}^{(i)}$ from $K_t^{(i)}$ and $k_t^{(i)}(\boldsymbol{x}_{t+1}) = K^{(i)}(\boldsymbol{x}_{t+1}, \boldsymbol{x}_j)$, $j = 1, \ldots, t$, as

    $$K_{t+1}^{(i)} = \begin{bmatrix} K_t^{(i)} & k_t^{(i)}(\boldsymbol{x}_{t+1}) \\ k_t^{(i)\top}(\boldsymbol{x}_{t+1}) & K^{(i)}(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1}) \end{bmatrix}$$

  Notice that the three sub-steps above can be performed in parallel across particles, with considerable gain in terms of computational speed.

The correlation range parameters and the latent input could be deterministically propagated by copying them from $S_t^{\zeta(i)}$ to $S_{t+1}^{(i)}$ since they do not change in t. Although this strategy is fast, it could lead to particle depletion in future resampling steps. To avoid degeneracy, we include a "*rejuvenate*" step which applies Markov Chain Monte Carlo (MCMC) moves to the particles after the propagating step Gilks and Berzuini (2001); Ridgeway and Madigan (2003). The update is done via elliptical slice sampling Murray et al. (2010).

We remark that each particle returns an estimate of predictive mean surface, $\hat{f}^{(i)}$, and predictive standard deviation, $\hat{\sigma}^{(i)}$. Likely, some of these particles will provide higher fidelity surfaces than others. We will take the average of the point-wise predictive distribution for each of the particles, the posterior mean predictive curve, as our prediction of f at new inputs

$$\hat{f} = E(f \mid S^{(i)}) = \frac{1}{N} \sum_{i=1}^{N} \hat{f}^{(i)}, \tag{5.6}$$

whereas the estimate for the predictive standard deviation is obtained as

$$\hat{\sigma} = \left\{ E[(\hat{\sigma}_i^2)_{i=1}^N] + \text{var}[(\hat{f}^{(i)})_{i=1}^N] \right\}^{\frac{1}{2}}. \tag{5.7}$$

We are currently working on an efficient `C++` implementation of the particle learning algorithm to be used in an `R` package. The prototype `R` code is available upon request.

Several authors have developed specific criteria for sequentially selecting new input points. For instance, Jones et al. (1998) proposed an expected improvement criterion to estimate the global minimum of a computer simulator via the maximum likelihood estimator for the emulator parameters. Equivalently popular approaches are the so-called active learning criteria such as active learning MacKay (MacKay,

1992) and active learning Cohn (Cohn, 1996). Seo et al. (2000) compared active learning MacKay and active learning Cohn and observed that active learning Cohn often performs better than active learning MacKay. For example, the active learning MacKay criterion embedded into a stationary Gaussian process emulator favors the selection of new points along the boundary of the input space in that the predictive variance is largest beyond the points which are already in the design (MacKay, 1992). However, the active learning Cohn criterion is more intensive to implement, therefore we will adopt active learning MacKay in our numerical examples for computational feasibility.

Active learning MacKay-based selection of future inputs sits comfortably within our particle learning implementation. After particles have been resampled, the algorithm performs prediction at a set of candidate input configurations based on the posterior predictive distribution (see Gramacy and Polson (2011) for more details). Active learning MacKay induces an ordering among candidate points based on their predictive standard deviation and the point with largest standard deviation in predicted output is chosen as the next input $\boldsymbol{x}_{t+1}$. Consequently, particles are propagated with the new pair $[\boldsymbol{x}_{t+1}, f(\boldsymbol{x}_{t+1})]$, and the sequence is iterated until some pre-specified stopping criterion is met, e.g. the largest predictive standard deviation falls below a certain threshold or a total number, T, of points has been included in the design.

5.3 Case studies

5.3.1 Learning local features

We consider a spatially inhomogeneous smooth function:

$$f(x) = \sin(x) + 2\exp(-30x^2), \tag{5.8}$$

which is evaluated at 15 equally spaced points in $\Omega = [-2, 2]$.

For particle learning, we use $N = 1000$ particles initialized at time $t_0 = 4$ with a randomly selected subset of size 4 of the original 15 points. $\{\phi_1, \phi_2\}$ and $\tilde{\phi}_1$ are assigned log-normal priors distributions, and 0.5 and 0.25 are chosen as the prior mean and prior variance of the corresponding Normal distribution on $\{\log \phi_1, \log \phi_2, \log \tilde{\phi}_1\}$. Also, a rather uninformative inverse-gamma prior is chosen for σ^2, $\sigma^2 \sim \mathrm{IG}(2, 1)$.

Figure 5.1 shows the posterior mean predictive curve together with error bars computed as $\hat{f} \pm 2\hat{\sigma}$. We also show the results of fitting Bayesian treed Gaussian process (Gramacy and Lee, 2008) and composite Gaussian process (Ba and Joseph, 2012) models. The limitations resulting from fitting a stationary Gaussian process to function (5.8) were outlined in Section 1. In comparison, the three non-stationary emulators (panels 2-4 in Figure 5.1) give significantly improved performance, i.e. the spline tension effect is eliminated, or strongly attenuated. However, treed Gaussian process' most evident feature is the large uncertainty in the estimates as quantified by very wide error bars, which could be taken as indicator of an inadequate representation of the simulator. The error bars obtained with our non-stationary Gaussian process and composite Gaussian process are more consistent with the local variability of the underlying surface. In terms of root mean squared error, our emulator improves the accuracy of treed Gaussian process and composite Gaussian process by 25% and 40%, respectively.

5.3.2 Quantifying the emulator's uncertainty

The simulator is typically expected to be within two or three standard deviations from the predictive mean (Bastos and O'Hagan, 2009). While an isolated outlier might be ignored, several large standardized residuals, e.g. more than 1% or 5% of the total number of validating points, may denote a problem to be further investigated. For example, large standardized residuals systematically observed in

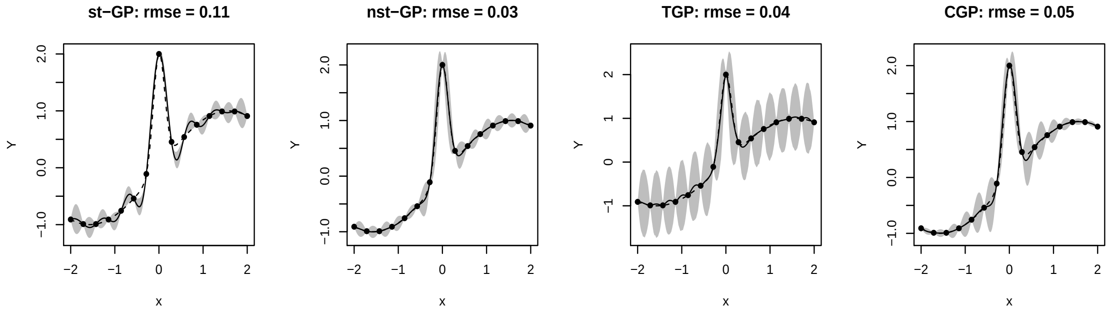


FIGURE 5.1: Comparison among stationary Gaussian process (st-GP), non-stationary Gaussian process via latent input augmentation (nst-GP), treed Gaussian process (TGP), and composite Gaussian process (CGP). Estimates (and root mean squared error – RMSE) are obtained at 200 equally spaced test points. The dashed line corresponds to the true function (5.8), the solid black line is the posterior mean predictive curve, and grey areas denote the error bars.

correspondence of a particular input suggest that the emulator is not learning the local behavior of the process (Busby, 2009). Further, they indicate that the emulator is under-estimating the predictive uncertainty. Ultimately, one wants to acquire an accurate knowledge of f with as least simulator's runs as possible. The emulator can be used to quickly identify those regions of the input space where the simulator exhibits more variations, thus help determine where the simulator's runs should concentrate. However, this goal can be achieved only if the emulator's estimate of uncertainty is trustworthy. If not, a sequential design strategy based on uncertainty will lead to a sub-optimal selection of input points.

Here we examine how model-based evaluations (Figure 5.1 and first row in Figure 5.2) combine with extrinsic diagnostics (second row in Figure 5.2). According to the exploration-driven predictive standard deviation of a stationary Gaussian process, one is basically equally likely to locate the new point anywhere in $[-2, 2]$ (first panel in Figure 5.2). Instead, cross validation strongly favors the selection of a new input around $x = 0$ (exploitation-driven cross validation) to learn the local behavior

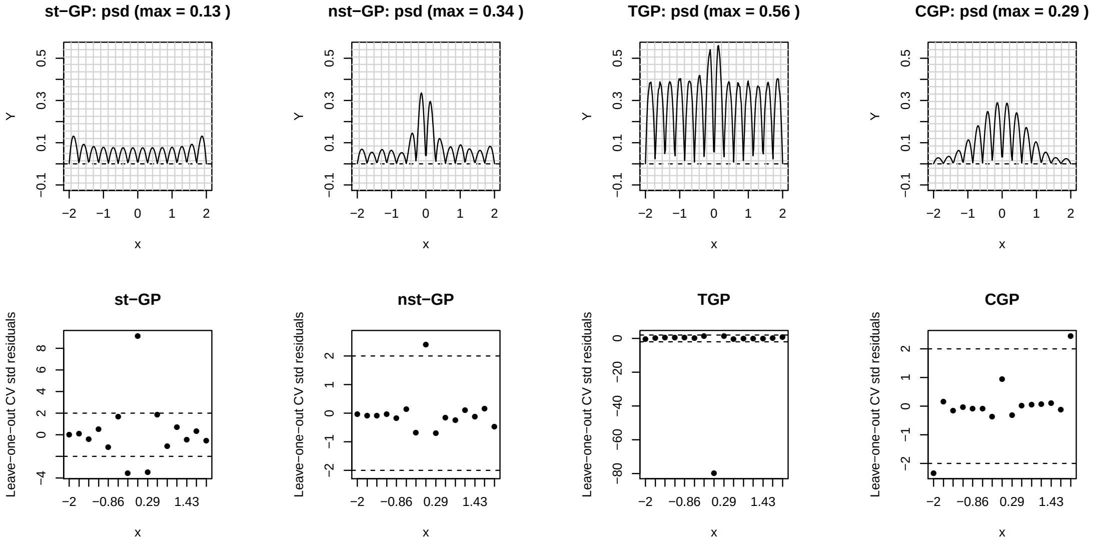


FIGURE 5.2: Predictive standard deviation (psd) at 200 predictive locations and leave-one-out cross validated (CV) standardized residuals for the peak function: comparison among stationary Gaussian process (st-GP), non-stationary Gaussian process via latent input (nst-GP), treed Gaussian process (TGP), and composite Gaussian process (CGP).

of f. Thus, model-based evaluations and extrinsic diagnostics are inconsistent, and the latter shows that uncertainty is being under-estimated around the peak. Incongruent conclusions with composite Gaussian process: if one trusts the model-based estimate of uncertainty, then the next input will be chosen around the peak; if one relies on cross validation, the next input will be chosen at the boundaries of the input space. For these two emulators, the problem of how to combine different diagnostic results emerges clearly. Instead, both model-based evaluations and cross validation for our non-stationary Gaussian process and treed Gaussian process identify that the next point is needed around $x = 0$. As opposed to our emulator, the conclusion with treed Gaussian process is however made much more evident by extrinsic diagnostics (a strikingly large cross validated standardized residual at $x = 0$) rather than by the predictive standard deviation.

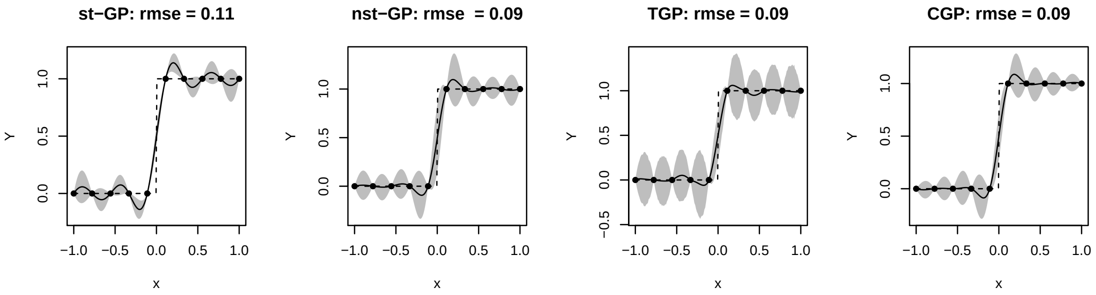


FIGURE 5.3: Plot of function (5.9) and estimates obtained with stationary Gaussian process (st-GP), non-stationary Gaussian process via latent input augmentation (nst-GP), treed Gaussian process (TGP), and composite Gaussian process (CGP).

5.3.3 1D discontinuous function

We now consider a simple discontinuous function:

$$f(x) = \begin{cases} 0, & x \leqslant 0 \\ 1, & x > 0 \end{cases} \tag{5.9}$$

with $x \in [-1, 1]$. We evaluate (5.9) at 10 equally spaced points in $\Omega = [-1, 1]$, and the initial design for particle learning is a randomly selected subset of size 4 of the same grid of points. This function is particularly suited to treed Gaussian process because of the vertical, axis-aligned nature of localized feature.

As opposed to the stationary Gaussian process, the point predictions made by the three non-stationary emulators are not (or less) distorted by the spline tension effect (Figure 5.3), and the intervals seem more consistent with what might be guessed about the function from observing the data points. Again, treed Gaussian process identifies large uncertainty everywhere in Ω.

Model-base evaluations for our non-stationary Gaussian process and composite Gaussian process suggest to pick new points at (and around) $x = 0$ to exploit the local feature (top row in Figure 5.4). Extrinsic and model-based evaluations are still inconsistent for the stationary Gaussian process, whereas extrinsic diagnostics show

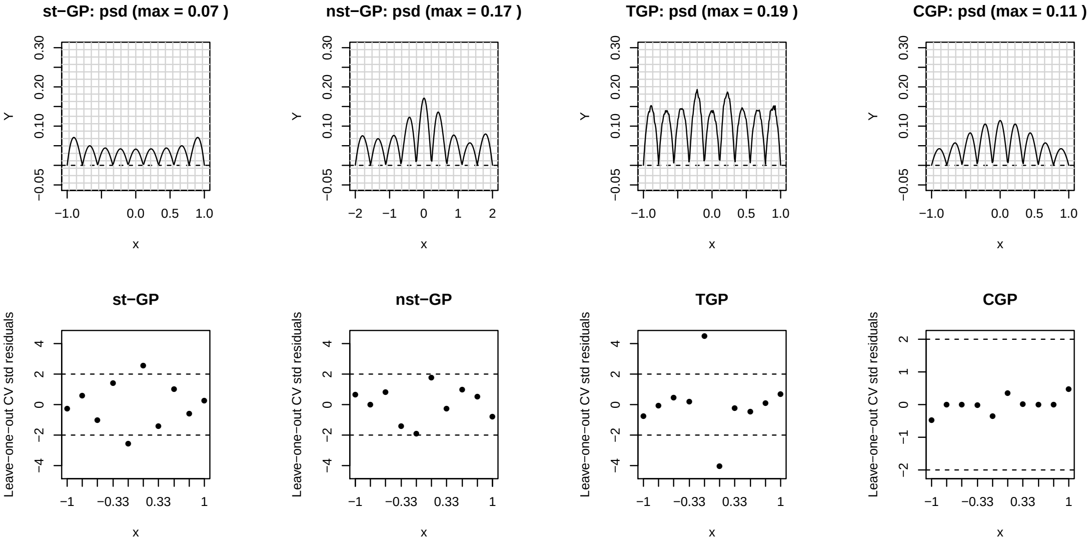


FIGURE 5.4: Predictive standard deviation (psd) at 200 predictive locations and leave-one-out cross validated (CV) standardized residuals for function (5.9): comparison among stationary Gaussian process (st-GP), non-stationary Gaussian process via latent input (nst-GP), treed Gaussian process (TGP), and composite Gaussian process (CGP).

that treed Gaussian process is likely to be under-estimating the uncertainty at the jump. Although this would lead to a sequential design strategy consistent with the one suggested by treed Gaussian process' model-based evaluation, it is preferable to observe the more apparent pattern in our emulators's predictive standard deviation, which drops quickly when departing from $x = 0$.

In general, it is not clear how to reconcile model-based and extrinsic diagnostics whenever these lead to different evaluations. In particular, it is not obvious in what measure to favor the exploration-driven predictive standard deviation over the exploitation-driven cross validation. An emulator whose model-based evaluations reconcile with extrinsic diagnostics is preferred in that it automatically learns to create a good balance between exploration and exploitation, and one does not have to resort to ad-hoc combinations. Our emulator seems to accomplish this balance

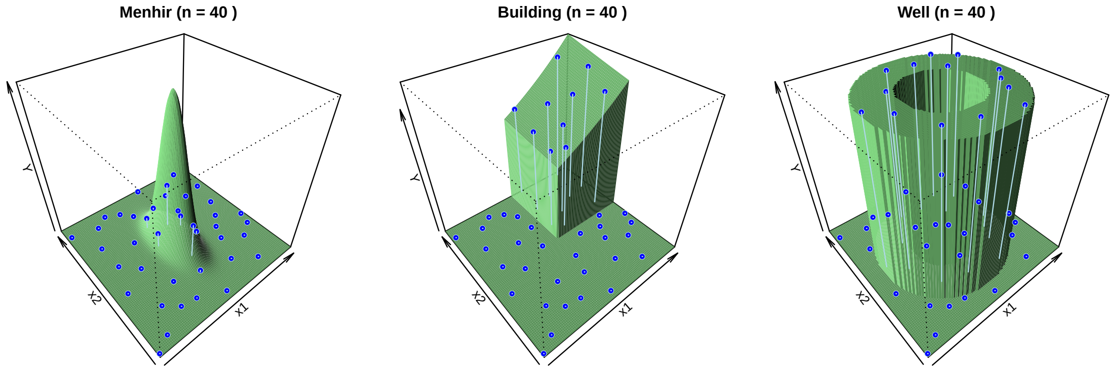


FIGURE 5.5: True functions for the 2-dimensional numerical examples.

adequately.

5.4 High-dimensional examples and sequential design

5.4.1 Two-dimensional functions with local features

In this Section, we provide three test functions possessing non-stationary features (Figure 5.5). The second function (building) is naturally suited to treed Gaussian process because of the axis-aligned non-stationarity.

First, we compare the performance of the emulators trained on the same set of input points. We use a 40 latin hypercube design (blue points in Figure 5.5), which allows the emulators gather knowledge on the overall shape of f because of its space-filling nature. Figure B.1 in Web Appendix B and Figs. 5.6-5.7 show the posterior predictive mean surface, $\hat{f}$, and the predictive standard deviation, $\hat{\sigma}$, for the menhir, building, and well functions, respectively. For the menhir function, the initial latin hypercube design does not include points at or nearby the peak, and this affects the estimates of the four emulators which can not recover the central spike. Note, however, how both our emulator and composite Gaussian process identify higher uncertainty in the central part of the input space. This is also true for treed Gaussian process, but the reason is likely to be related to the partitioning scheme rather than

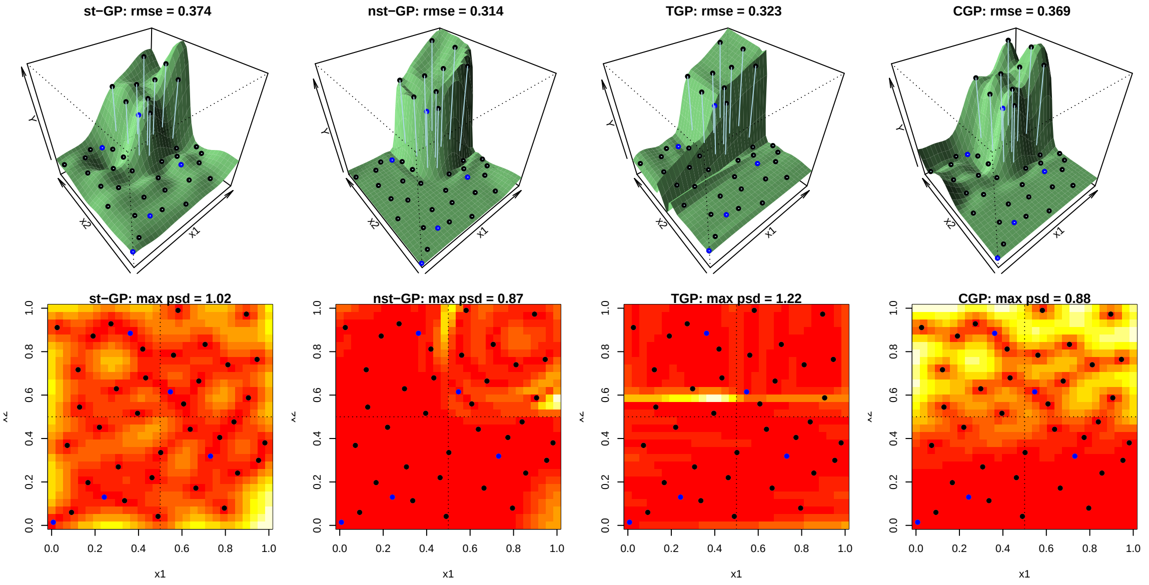


FIGURE 5.6: Building function estimates at $T = 40$. The quality of the prediction is assessed at a collection of 900 points in $[0, 1]^2$, i.e. an expanded grid of 30 equally spaced points along each coordinate axes. Root mean squared error (RMSE) and maximum predictive standard deviation (max psd) based on the test points are also reported. Blue points are a randomly selected subset of the latin hypercube design used for initialization of particle learning. Comparison among stationary Gaussian process (st-GP), non-stationary Gaussian process via latent input (nst-GP), treed Gaussian process (TGP), and composite Gaussian process (CGP).

to learning the geometry of the feature. The most distinctive feature that emerges from both Figure 5.6 and Figure 5.7 is that our emulator is learning the geometry of the local features, as shown by the evident patterns in predictive standard deviation. This does not appear to be the case for the other emulators.

Next, we want to assess whether the emulators can correct for inadequacies in the fit. In other terms, we want to examine whether the emulators can learn about, and thus concentrate exploration in, the most interesting or complicated regions of the input space. Therefore, we let the emulators select 20 additional points (60 for well) sequentially via active learning MacKay.

Figures B.2 and B.3 in Appendix B show $\hat{f}$ and $\hat{\sigma}$ for the menhir and well func-

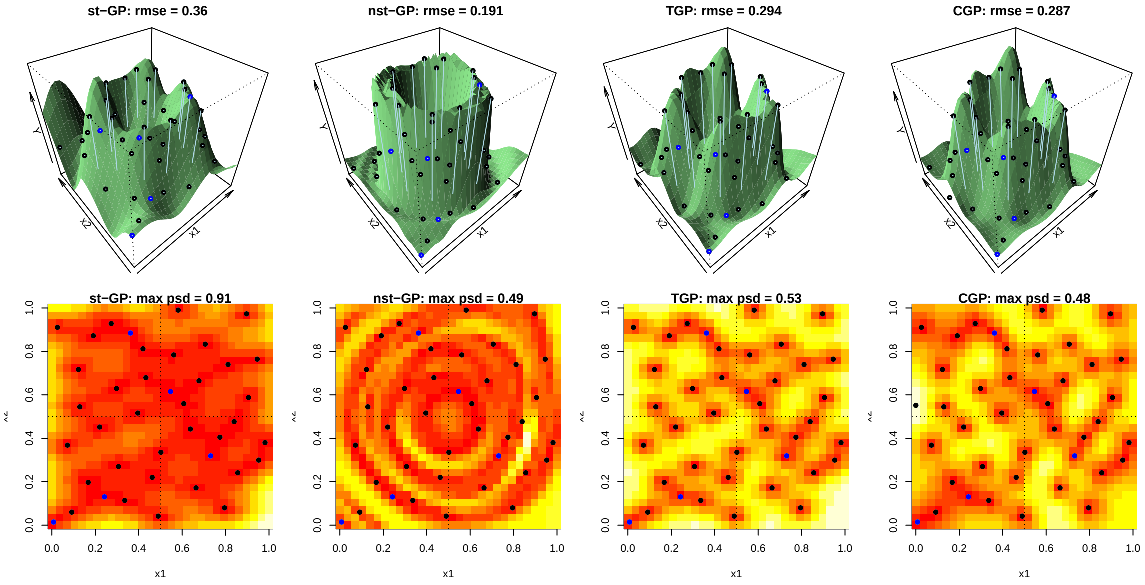


FIGURE 5.7: Well function estimates at $T = 40$. The quality of the prediction is assessed at a collection of 900 points in $[0, 1]^2$, i.e. an expanded grid of 30 equally spaced points along each coordinate axes. Root mean squared error (RMSE) and maximum predictive standard deviation (max psd) based on the test points are also reported. Blue points are a randomly selected subset of the latin hypercube design used for initialization of particle learning. Comparison among stationary Gaussian process (st-GP), non-stationary Gaussian process via latent input (nst-GP), treed Gaussian process (TGP), and composite Gaussian process (CGP).

tions at $T = 60$ and $T = 100$, respectively, and Figure 5.8 refers to the building function at $T = 60$. Regardless of the function being examined, our non-stationary emulator favors the sampling of new points from the boundaries of the features. Therefore, it strikes a good balance between exploration (initial latin hypercube design) and exploitation (newly selected points). This is not necessarily true for the other emulators across different functions, i.e. composite Gaussian process tends to select new points at the center of the input space of the menhir function, but no pattern is observed for building and well functions. Treed Gaussian process' selection is driven by the partitioning scheme in that the predictive standard deviation is generally higher at the edges between consecutive partitions. Thus, treed Gaussian

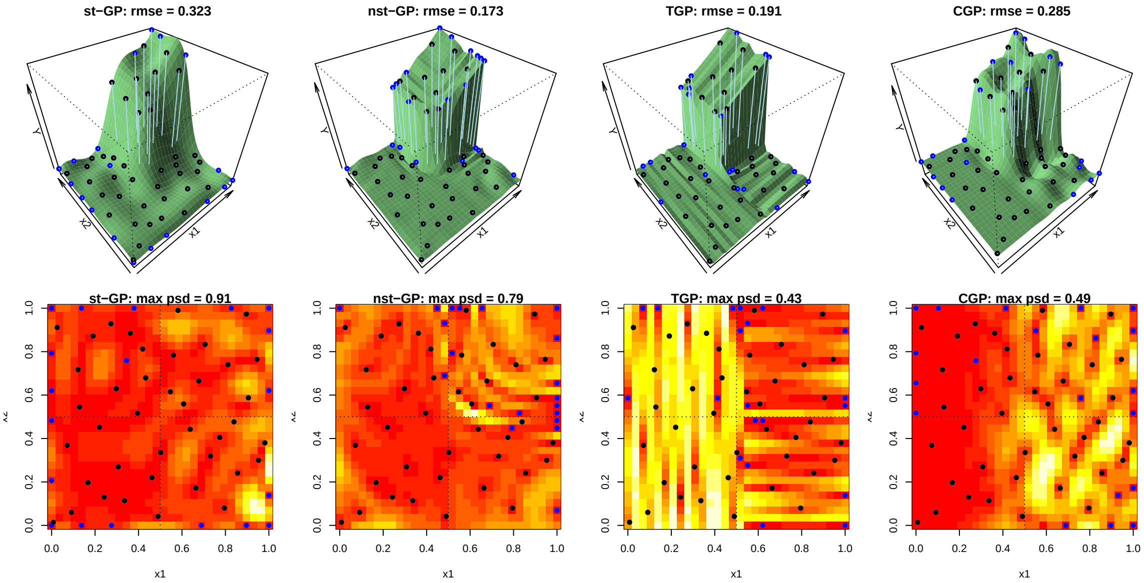


FIGURE 5.8: Building function estimates at 60 design points. Blue points denote the additional inputs selected via active learning MacKay. Comparison among stationary Gaussian process (st-GP), non-stationary Gaussian process via latent input (nst-GP), treed Gaussian process (TGP), and composite Gaussian process (CGP).

process seems to concentrate in learning the partition rather than the local feature.

For a more quantitative numerical comparison among the emulators, Figure 5.9 shows the progression of the root mean squared error as additional inputs are being selected. Our non-stationary emulator performs at least as well as composite Gaussian process on the menhir function, and outperforms the other emulators on building and, in particular, well functions.

To conclude, our emulator is learning and concentrating the exploration in interesting areas of the input space. Furthermore, it compares favorably both in cases of axis-aligned non-stationarity (building) and in situations where the type of non-stationarity is more general (well).

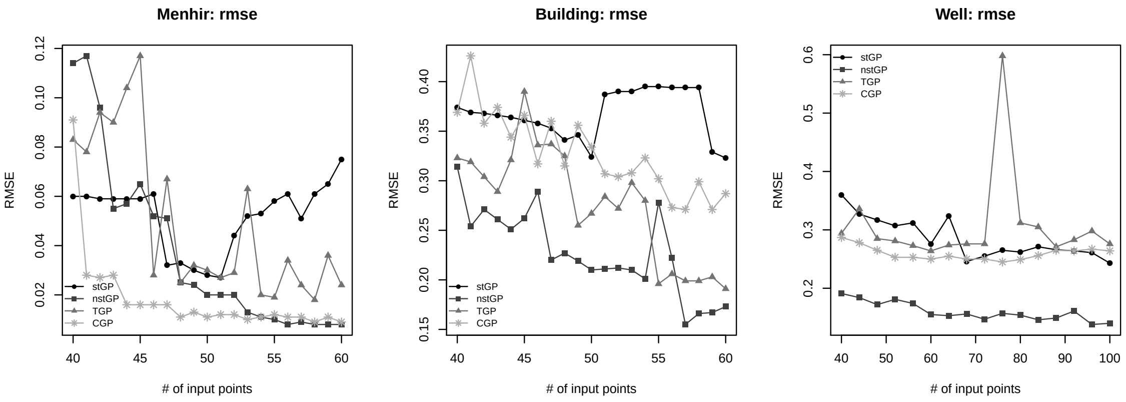


FIGURE 5.9: Progression of the root mean squared error (RMSE) as additional input points are being selected for the 2D functions. Comparison among stationary Gaussian process (stGP), non-stationary Gaussian process via latent input (nstGP), treed Gaussian process (TGP), and composite Gaussian process (CGP).

5.4.2 Six-dimensional examples

We consider two 6D examples, which constitute an extension of the 2D building and well functions. The 6D building has true function:

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = \begin{cases} e^{\sum_{i=1}^{6}\left(\frac{1}{i}\right)^2 x_i}, & \text{if } x_1, x_2, x_3, x_4, x_5, x_6 > 0.25 \\ 0, & \text{otherwise} \end{cases} \tag{5.10}$$

on the hypercube $X = [0, 1]^6$. The 6D well has true function:

$$f(x_1, \ldots, x_6) = \begin{cases} 1, & \text{if } \sum_{i=1}^{4}(x_i - 0.5)^2 > 0.025 \text{ and } \sum_{i=1}^{4}(x_i - 0.5)^2 < 0.25 \\ 0, & \text{otherwise} \end{cases} \tag{5.11}$$

on the hypercube $X = [0, 1]^6$. Therefore, f in (5.11) is constant in x_5 and x_6.

For particle learning, $N = 1000$ particles are trained on a 120 latin hypercube design. Emulators then select 80 additional points from a 1000 candidate latin hypercube design according to active learning MacKay. Similar to the 2D examples, our non-stationary emulator outperforms the others in terms of reduction of the

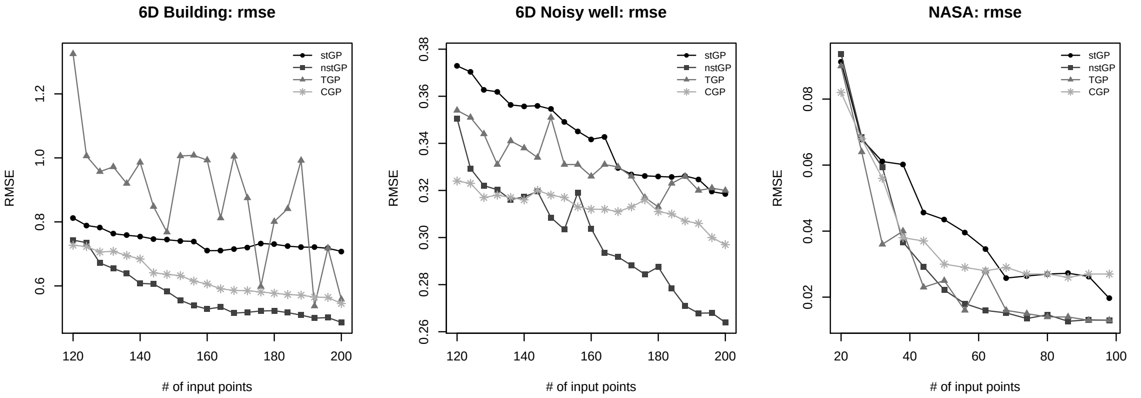


FIGURE 5.10: Progression of the root mean squared error (RMSE) as additional input points are being selected. Comparison among stationary Gaussian process (stGP), non-stationary Gaussian process via latent input augmentation (nstGP), treed Gaussian process (TGP), and composite Gaussian process (CGP). Left and central panels: 6D examples; Right panel: Langley glide-back booster experiment.

root mean squared error (left and central panels in Figure 5.10). Therefore, inactive covariates which add noise to the process (6D well) do not affect the performance of our emulator. Additional summaries are reported in Appendix B.

5.5 Langley glide-back booster experiment

This Section presents an application to a computational fluid dynamics simulator of a proposed reusable NASA rocket booster vehicle, the Langley glide-back booster. The interest is in learning about the response in several flight characteristics of the Langley glide-back booster as a function of three inputs (speed in Mach number, angle of attack, and slide-slip angle) when the vehicle reenters the atmosphere. See Gramacy and Lee (2009) for more details on the study.

The computational fluid dynamics simulation involves the iterative integration of systems of inviscid Euler equations and each run of the solver for a given set of parameters takes on the order of 5–20 hours on a high-end workstation (Gramacy and Lee, 2009). Therefore, the interest in adaptively design the experiment to concentrate sampling in those regions where the response is more interesting (e.g., higher uncertainty or richest structure) emerges clearly. As Gramacy and Lee (2009) show, the most interesting region occurs near Mach 1 and for large angle of attack (refer to Figure B.6 in Appendix B which shows the lift response as function of Mach and Alpha). The ridge in response at Mach equal to 1 separates subsonic flows and supersonic flows. The behavior of the response is quite different in the two regions, with lift appearing mostly homogeneous in the supersonic region.

Following Gramacy and Lee (2009), we examine the lift response as a function of speed (Mach) and angle of attach (Alpha) with the side-slip angle (Beta) fixed at zero. We obtain a linear interpolation onto a 30×30 grid over Mach and Alpha, and use the interpolated lift as our truth. Figure 5.11 shows a slice of the posterior mean predictive surface as a function of Mach and Alpha. The distinction between subsonic and supersonic flows is well captured by the non-stationary emulators, which tend to select new input points with small Mach, particularly for large Alpha. The stationary Gaussian process focuses mostly on a uniform exploration of the space and will require ad-hoc extrinsic diagnostics to focus around the ridge.

The third panel in Figure 5.10 shows the progression of the root mean squared error to the interpolated truth. Our emulator performs as well as treed Gaussian process on a surface that favors the latter because of the axis-aligned local feature, and improves the accuracy over stationary Gaussian process and composite Gaussian process by 35% and 48%, respectively, at $T = 100$.

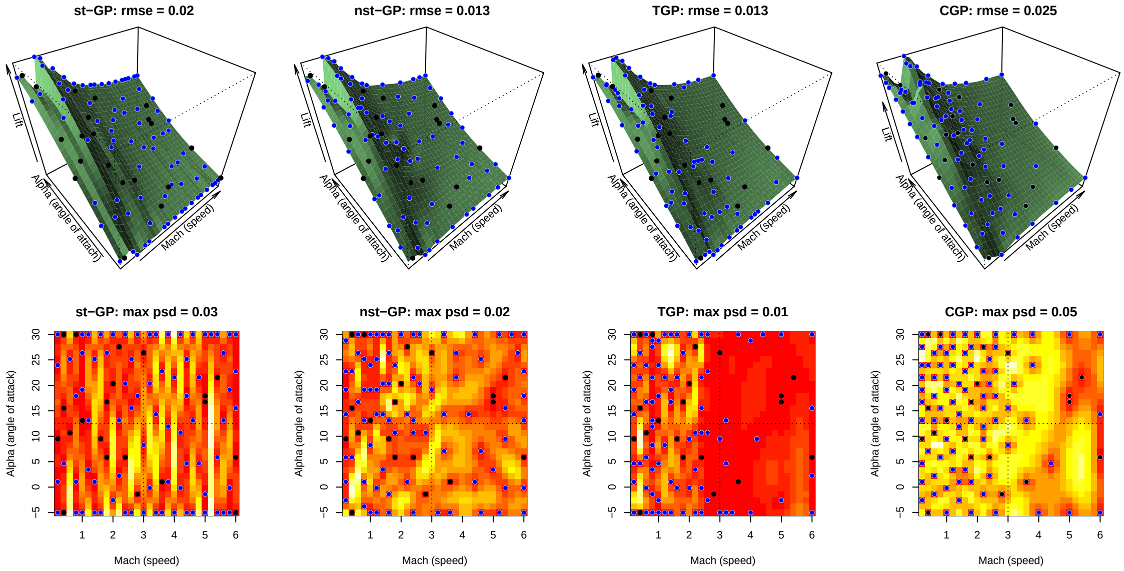


FIGURE 5.11: Langley glide-back booster slice of mean posterior predictive surface of the lift response as a function of Mach (speed) and Alpha (angle of attack) with Beta (side-slip angle) fixed at 0. The design was initialized with 20 randomly selected points from a 30 × 30 grid (black points), and 80 new points were selected via active learning MacKay (blue). Comparison among stationary Gaussian process (st-GP), non-stationary Gaussian process via latent input augmentation (nst-GP), treed Gaussian process (TGP), and composite Gaussian process (CGP).

5.6 Two-stage empirical Bayes approximation of non-stationary Gaussian process emulator

5.6.1 Implementation

The modeling approach we have investigated in the previous Sections is formulated around an effort at joint modeling of both f and Z. The latent input Gaussian process essentially becomes a vector of model parameters that can not be marginalized out within our construction, thus needs to be learnt. Bearing in mind that this Gaussian process is latent and the vector Z is of sequentially increasing dimension, the learning of Z can face challenges and a large amount of data may be needed to learn it well. Alternatively to this full Bayes approach, we elect here a cruder but much simpler strategy to approximate our non-stationary emulator. Specifically, we can rely on a two stage approach where the first stage focuses on the estimation of the latent predictor, and the second stage focuses on learning f assuming that the latent predictor is known and fixed at the level estimated in the first stage. Although several methods exist to obtain an estimate of the latent input Gaussian process at first stage, we investigate here an Markov chain Monte Carlo-based nonparametric regression approach.

Suppose we are given an initial design $\{\boldsymbol{x}_i, f(\boldsymbol{x}_i)\}_{i=1}^{t}$. We consider $Z = g(\boldsymbol{x})$ and model g by a stationary Gaussian process indexed by the p-dimensional vector of known inputs, $\boldsymbol{x}$, and a vector of model parameters, $\boldsymbol{\theta}$:

$$g \mid \boldsymbol{\theta} \sim \mathrm{GP}(\tilde{\mu}_{\boldsymbol{\theta}}, \tilde{K}_{\boldsymbol{\theta}}). \tag{5.12}$$

As discussed above, several choices are available for the Gaussian process mean $\tilde{\mu}_{\boldsymbol{\theta}}$, including the constant-zero mean. The correlation function $\tilde{K}_{\boldsymbol{\theta}}$ corresponds to (5.5). If we consider a unit scale and fix $\tilde{\mu}_{\boldsymbol{\theta}} \equiv 0$, (5.12) reduces to the Gaussian process prior on g presented in (5.3). One can estimate g from a smooth Gaussian process (noisy) regression:

$$f(\boldsymbol{x}_i) = g(\boldsymbol{x}_i) + \epsilon_i, \quad \text{with} \quad \epsilon_i \sim \mathrm{N}(0, \tau^2), \quad \text{and} \quad i = 1, \ldots, t. \tag{5.13}$$

Overall, the model in (5.12)-(5.13) is equivalent to assuming a Gaussian process prior on f:

$$f \mid \boldsymbol{\theta}, \tau^2 \sim \mathrm{GP}(\tilde{\mu}_{\boldsymbol{\theta}}, \tilde{K}_{\boldsymbol{\theta}} + \tau^2 \delta_{j,k}), \tag{5.14}$$

where $\delta_{\cdot,\cdot}$ is the Kronecker delta function. Bayesian inference of (5.14) proceeds via Markov chain Monte Carlo: realizations are drawn from the joint posterior distribution of the model parameters and, for all $\boldsymbol{x}$ of interest and using the parameter values drawn from the posterior distribution, we can estimate Z at $\boldsymbol{x}$ as $\hat{g}(\boldsymbol{x}) = E[f(\boldsymbol{x}) \mid \boldsymbol{x}, \boldsymbol{\theta}, \tau^2]$, the point predictor of f at $\boldsymbol{x}$. Steps are repeated a large number of times, and the average of the point predictors is used as estimate of g. At the second stage, we consider $f \mid \hat{g}(\boldsymbol{x}), \boldsymbol{\theta} \sim \text{GP}(\mu_{\boldsymbol{\theta}}, K_{\boldsymbol{\theta}})$, where f is now a stationary Gaussian process indexed by a $p+1-$dimensional vector of inputs $\{\boldsymbol{x}, \hat{g}(\boldsymbol{x})\}$ as in (5.3)-(5.4) under the fiction that the latent input is known. Similar to the first stage, inference for $\boldsymbol{\theta}$ proceeds via Markov chain Monte Carlo. Similar to the full Bayes version of our method, prediction is made at a set of candidate points using the the parameter values drawn from the posterior distributions, and the estimated predictive uncertainty is used to guide the selection of new inputs.

The two stage, Markov chain Monte Carlo-based inference is perfectly coherent and comes closest to a full Bayesian treatment of the problem since it takes into account uncertainty in estimating the hyperparameters and the latent input Gaussian process at first stage. However, Markov chain Monte Carlo-based inference is ill-suited to sequential design, as the chain must be restarted and iterated until convergence when the design is augmented with a new pair $[\boldsymbol{x}_{t+1}, f(\boldsymbol{x}_{t+1})]$. Fits from previous iterations can only guide the initialization of the new Markov Chain.

5.6.2 Simulation studies

We evaluate the performance of the two stage approximation on the sequential experiments presented in the previous Sections. Figure B.7 in Appendix B shows the performance of the two stage approximation on the 2D examples of Section 5.4.1 given the initial 40 latin hypercube design. Estimating the latent input surface at first stage considerably improves the performance of our emulator on the Menhir function. Similar to what observed with the full Bayes implementation, the key feature is that the emulator is learning the geometry of the different features. This helps select new inputs from the edges of such features (Figure 5.12). Figure 5.13 shows a comparison between two stage approximation and full Bayes in terms of progression of the root mean squared error. No implementation is preferred in terms of predictive accuracy across functions or number of input points. The full Bayes approach is preferred on the well function, whereas two stage seems to be preferred on the Menhir function for smaller designs. This is probably due to the ability of the two stage approximation in better learning the latent input Gaussian process through the first noisy regression, which results into quicker learning of f. However, the predictive accuracy of the full Bayes approximation of our emulator considerably improves when one point is selected at the center of the input space (this happens at $t = 48$), and eventually reconciles with two stage approximation. Figure B.8 in Appendix B shows that the full version of our emulator outperforms the two stage approximation on the 6D and NASA experiments.

To conclude, the two stage approximation of our emulator preserves some good features of the full Bayes version, namely learning the geometry of different types of shape and increasing the sampling frequency of new inputs along important input dimensions. Therefore, it constitutes a valid alternative to the full Bayesian implementation for adaptive design selection and function approximation. However, the full Bayes version often achieves lower root mean squared error, in particular for larger designs or in higher dimensions.

5.7 Discussion

In this work we describe a non-stationary Gaussian process model that can be used as an emulator in the sequential design of computer experiments. To induce non-stationarity, we consider a mapping to a latent space where stationarity holds, and augment the input space by the latent input. The numerical examples show that the extra flexibility introduced by the latent input greatly improves predictions over a stationary Gaussian process fit. In particular, the proposed methodology provides more reliable, model-based evaluations as opposed to extraneous explorations done with stationary Gaussian processes, and adapts to both cases of axis-aligned non-stationarity and in situations where the non-stationarity is more general. The

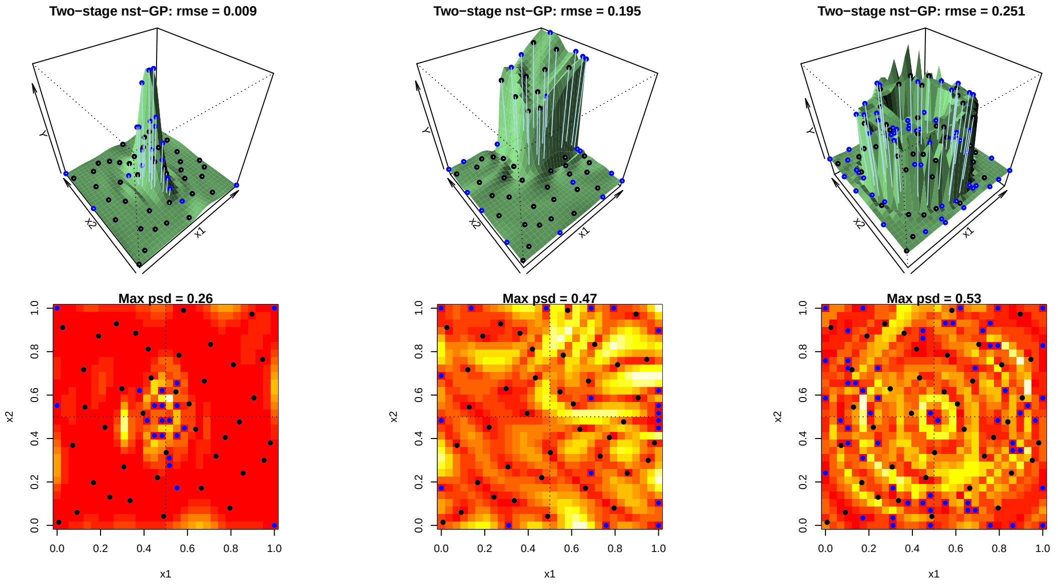


Figure 5.12: Two stage approximation: the quality of the prediction is assessed at a collection of 900 points in $[0,1]^2$, i.e. an expanded grid of 30 equally spaced points along each coordinate axes. Blue points are additional points selected via active learning MacKay criterion. The plot also reports the root mean square error (RMSE) and the maximum predictive standard deviation (max psd) computed at the test set.

approach also retains an easy interpretability while building upon a simple but elegant construction. Here we discuss some details in regard to our implementation and computer emulation in general.

The nugget. A nugget is a small, positive quantity α often added to the diagonal of the correlation function for f (Andrianakis and Challenor, 2012). The resulting covariance function corresponds to the case where f is observed with additive Gaussian noise with zero mean and variance α. Many authors do not include a nugget term on the grounds that computer codes are deterministic. In fact, the nugget introduces a measurement error in the stochastic process. A Gaussian process that includes a nugget does not interpolate and assigns non-zero uncertainty to the design data. However, it is not uncommon practice to include a nugget to enhance the numerical stability in factorizing covariance matrices (Gramacy and Lee, 2008;

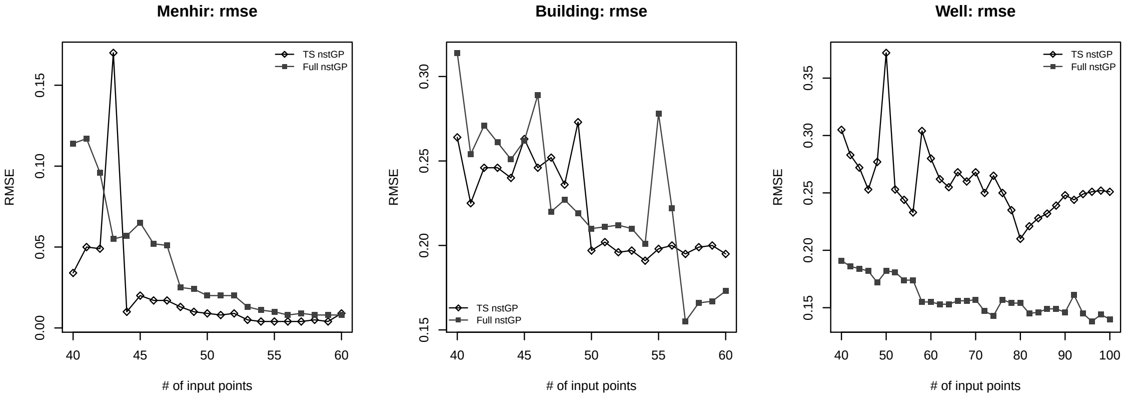


Figure 5.13: Two stage fast approximation: progression of the root mean squared error (RMSE) as additional input points are being selected for the 2D functions. Comparison between the two stage version of the non-stationary Gaussian process (TS nstGP) and the full Bayes approach implemented via particle learning (Full nstGP).

Andrianakis and Challenor, 2012). A typical value of the nugget used in our numerical examples is $\alpha = 10^{-7}$ (particle learning implementation), the effect of this being the addition of α in the predictive variance of the responses. Although very small, α can have a non-negligible impact on the estimates. For example, it compromises interpolation of the stationary Gaussian process on the menhir function (Figure B.2 in Appendix B). However, the nugget did not seem to significantly affect the estimates of our non-stationary Gaussian process. For more details on the inclusion of a nugget in computer emulation, refer to Andrianakis and Challenor (2012); Ba and Joseph (2012).

High-dimensional problems. The application of the proposed methodology to high-dimensional input spaces can be challenging due to the intrinsic difficulty faced in high-dimensional settings by Gaussian process models, which try to recover up to the *p-th* level of interaction. We expect that more structure (i.e., additivity, sparse factorization) is needed to handle high dimensional problems. Independently developed research in the context on non-parametric regression suggests that additive Gaussian process models could be a promising way to move forward, and they will be investigated in future research.

Applications. Although the model was developed for the analysis of computer experiments, it also has a wide range of uses as a simple and efficient method for non-stationary modeling in the analysis of social, biological, and ecological data collected over spatial domains. The extension to non-parametric regression is straightforward with the inclusion of a nugget (Schmidt and O'Hagan, 2000; Rasmussen and Ghahramani, 2001; Kim et al., 2005; Paciorek and Schervish, 2006).

6

Conclusions and future directions

This dissertation focused on Bayesian methodologies for functional data and computer emulators. In regard with functional regression, we explored the use of latent factor models as a vehicle to provide a low dimensional representation of the curves and allow joint modeling of functional predictors and scalar or vector-valued outcomes. Along the same line, latent factors could be used in context of function-on-function regression or to link multiple functions recorded on the same subject. The idea is similar to the structure developed in Chapter 4 to accommodate dependence across the time-course gene expression trajectories. For example, in longitudinal studies it is common to record several quantities of interest on the same subject over time. Ideally, one would want to build a full joint model where each measured variable is allowed to have its own trajectory while accommodating dependence among these trajectories. To accomplish this goal, the $\boldsymbol{\theta}_i$ vector of basis coefficients in the functional data model (Chapter 2) could instead be replaced with concatenated coefficients within component models and made dependent on a common set of subject-specific latent factors, thus effectively accommodating dependence through the trajectories. Models of this kind could be built for different types of objects, including not only time trajectories but also images, movies, text, etc. This would lead to a general shared latent factor framework for modeling high-dimensional mixed domain data that should have broad utility to be explored in future research.

An interesting area of application for a general shared latent factor framework is in the analysis of multivariate response data measured on diverse scales. This type of data is routinely collected in the social sciences and in other scientific fields, for example, in the form of surveys, questionnaires, etc. Generalized latent trait models have proven useful for joint modeling of variables having mixed dichotomous, categorical, continuous and count measurement scales. In these types of models, a separate generalized linear model (GLM) is defined for each measured variable. The different GLMs can correspond to different distributions in the exponential family and to different link functions, and dependence among the responses is accommodated by common latent variables in the linear predictors (Sammel et al., 1997; Bartholomew and Knott, 1999; Moustaki and Knott, 2000). Hence, these models generalize typical normal or probit factor models to a much broader class. However, there are two main practical issues that arise in the implementation. The first is that computation becomes daunting outside of the underlying normal family of models, so that it becomes infeasible to consider large numbers of measured variables and more than a small number of latent factors. The second is that there may be a lack of robustness to outliers and the parametric distribution of the latent variables (Bartholomew and Knott, 1999; Wedel and Kamakura, 2001; Noh and Lee, 2007). In regard with robustness to outliers, one can accommodate outliers in normal linear regression models by using error distributions that are heavy-tailed relative to the normal distribution (West, 1984). In particular, West (1984) focuses on a wide class of heavy-tailed, unimodal and symmetric error distributions that can be constructed as scale mixtures of normal distributions. This class of heavy-tailed prior distributions, which includes the t distribution, is particularly useful in a Bayesian context since it induces conditionally conjugate posterior distributions and leads to analytically tractable analysis. As for robustness to the parametric distribution of the latent factors, most of latent trait models rely on the assumption that the latent variables follow a normal distribution. Although this choice is reasonable in many applications, it may be too restrictive in other cases and not based on a scientific argument. Therefore, it is desirable to accommodate a wider class of distributions of the latent variables. For example, one could define flexible latent variable models using mixtures of parametric models.

The aforementioned issues will be investigated in future research. The goal is to propose an efficient Bayesian approach to posterior computation in a class of robust generalized linear latent variable models. Robustness to outliers can be incorporated through placing t-distributed residuals in the linear predictors, while robustness to the latent variable distributions can be accommodated through Dirichlet process mixtures. To enhance the update of the canonical parameters for binary and count data, which necessitates of a MH step within the Gibbs sampler, one could resort to weighted least squares proposal distributions as in Gamerman (1997). Such proposals mimic the true posterior distribution and should lead to an increase in the MH acceptance rate.

The second part of the dissertation focused on new methodologies for the statistical analysis of computer models. Perhaps one of the most important issues which has gone a bit under-explored in current research are the challenges that high-dimensional input spaces pose to an efficient estimation of the simulator. As a rule of thumb, the input space can be considered high-dimensional when its dimension is larger than 10. In some applications, the dimension of the input space, p, may be as large as 60 or above, however it is likely that the simulator output, f, is affected by a much smaller subset of these p inputs. The number of design points required to accurately estimate f generally increases exponentially in p. However, large designs contrast the very original idea of emulation itself, which is meant to provide a fast and accurate knowledge of f with a design as small as possible to save on costly-to-evaluate computer simulations. The very same Gaussian process (GP) emulation, and regression in general, is very suitable to the recovery of non-linear effects and interactions in low dimensions, but faces severe challenges in high dimensions. In particular, a single GP tries to recover an "all way" interaction, i.e. tries to select $d << p$ predictors which all interact in a single covariance function.

To overcome this issue, one could consider representations of f lowering the complexity of a high-dimensional space, e.g. additive models. In this regard, the emulator can be decomposed into a sum of low-dimensional functions, each depending only on a subset of the input variables. In our context, the component, low-dimensional functions can taken to be GPs. Additive GPs have been proposed in the context of regression by Duvenaud et al. (2011). The basic idea is that if there are $d << p$ important predictors, some of them may be main-effects (i.e. they affect the true function in a univariate way), while other important variables may have an interaction effect. An additive GP tries to split the learning of the regression surface into sub-components, allowing each GP to learn a different part of the signal, and in doing so, allowing the GPs to learn a complex signal in together. Additive GPs should provide a flexible enough structure to represent complex functions including those with non-stationary shapes examined in Chapter 5. However, any methodological advancement will need to be embedded into a smart design strategy and be implemented with scalable computational techniques that are suited to the online nature of sequential design.

Appendix A

Additional supporting material for the latent factor regression model

This appendix presents additional supporting material and plots that show the performance of the proposed latent factor regression model as described in Chapter 2 in a variety of examples.

Choosing the number of latent factors k adaptively

The number of latent factors, k, is tuned as the sampler progresses, with adaptations designed to satisfy the diminishing adaptation condition in Theorem 5 of Roberts and Rosenthal (2007). Following Bhattacharya and Dunson (2011a), we adapt with probability $p(t) = \exp\{\alpha_0 + \alpha_1 t\}$, with t denoting the t-th iteration and α_0, α_1 chosen so that adaptation occurs around every 10 iterations at the beginning of the chain and then decreases in frequency exponentially fast. In our application, we set $\alpha_0 = -1$ and $\alpha_1 = -5 \times 10^{-4}$. At every iteration, a random number u_t is sampled from a uniform distribution Unif(0,1), and adaptation occurs if $u_t \leqslant p(t)$. Whenever adaptation occurs, we count the columns of $\mathbf{\Lambda}$ having all elements in some pre-specified neighborhood of zero. We can intuitively assume that the factors corresponding to such columns have a negligible contribution, therefore we discard these columns of $\mathbf{\Lambda}$ and continue the sampler with a reduced number of factors, which also helps save computing time. Otherwise, if the number of such columns drops to zero we may be missing important factors, therefore we add a column to the loadings. The other parameters are modified accordingly and, when a factor is added, the new parameters are sampled from their prior distributions. Refer to Bhattacharya and Dunson (2011a) for further details on the adaptive Gibbs sampler.

Setting hyperparameters of the latent factor regression model

To facilitate the routine implementation of the proposed method, the Matlab codes for the LFRM and its joint modeling extensions (Section 2.5) are available at the *Biometrics* website on Wiley Online Library.

To implement our methodology, one has to choose the hyperparameters for the priors in Section 2.2 and the parameters ν in (2.3) and p in (2.2). Likely, the most daunting task is the choice of the bandwidth ν, that we fixed to 4 as a reasonable default value to ensure smooth trajectories. In general, one can not obtain curves bumpier than the resolution determined by the bandwidth, thus the choice of ν requires careful sensitivity analysis to identify a value which induces the desired level of smoothness for the trajectories. Applications with trajectories not having the same level of smoothness everywhere would require spatially adaptive smoothness, which can potentially be achieved by choosing a pre-specified finite dictionary of different bandwidths and then allowing the kernels to have varying unknown bandwidths via a griddy-Gibbs sampler.

The basis function representation in Equation 2.2 requires the choice of a truncation p. In general, one can include a rich, pre-specified set of basis functions ($p \approx 10, 20$ or larger) since the model allows automatic shrinkage and effective removal of basis coefficients not needed to characterize any of the curves under study, thus effectively induces basis selection. In our blood pressure application, the choice $p = 10$ ensured sufficiently many equally-spaced kernels to capture a high variety of smooth trajectory shapes.

Other parameters that need to be determined in the MCMC algorithm include υ, a_1, a_2 in (2.8)-(2.9). As remarked in Section 2.2, a choice $a_2 > 1$ induces stochastically increasing τ_h in (2.8), which favors more shrinkage as the column index increases. We set $\upsilon = 5$ and $a_1 = a_2 = 1.5$, but our sensitivity analyses showed robustness to different choices of these hyperparameters. Furthermore, one has to choose a_σ and b_σ, which are the inverse-gamma hyperparameters values for σ_j^2, and a_φ and b_φ, which are the inverse-gamma hyperparameter values for the measurement error variance φ^2. Our suggestion is to fix a mean and variance for the inverse-gamma priors and solve for the hyperparameters.

Alternatively to the Cauchy prior in (2.11), one could choose a Gaussian prior distribution for the $\boldsymbol{\beta}$ coefficients but this leads to poorer performance if a subsample of women has very sparse measurements. This occurrence is common when dealing with longitudinal data, which often consist of few and sparse measurements per subject, and it is verified in the blood pressure data where a group of women has few observations, usually located in the second half of the pregnancy. For this group of women, the prior becomes more influential and the intercept is pulled closer to zero than for women with more observations, resulting in an undesired low MAP trajectory estimate at early pregnancy.

As for the bivariate probit model in Section 2.4.2, we chose normal and multivariate normal priors for the additional model parameters. The prior for the intercept on the latent indicator of preeclampsia, α_1, was set to be $\alpha_1 \sim \mathrm{N}(\Phi^{-1}(0.12), 0.25)$, whereas the prior on the intercept for the latent indicator of low birth weight, α_2, was set to correspond to $\alpha_2 \sim \mathrm{N}(\Phi^{-1}(0.082), 0.25)$. The hyperprior mean for α_1 was set to be moderately high provided that the proportion of preeclamptic women in the sample is over twice the typical incidence range of 5-8%, and that of α_2 was chosen to correspond to the national average. Finally, $\boldsymbol{\gamma}_1 \sim \mathrm{N}_k(\boldsymbol{\mu}_{\gamma,1}, \boldsymbol{\Sigma}_{\gamma,1})$ and $\boldsymbol{\gamma}_2 \sim \mathrm{N}_k(\boldsymbol{\mu}_{\gamma,2}, \boldsymbol{\Sigma}_{\gamma,2})$, with $\boldsymbol{\mu}_{\gamma,1} = \boldsymbol{\mu}_{\gamma,2} = \mathbf{0}$, and $\boldsymbol{\Sigma}_{\gamma,1} = \boldsymbol{\Sigma}_{\gamma,2} = \mathbf{I}_k$. We repeated the analysis for a variety of these hyperparameter values (i.e., with the variance multiplied by 2 and divided by 2, etc.), but no noticeable differences were found in the results.

Finally, we examined the joint model of birth weight, gestational age at delivery and blood pressure. Specifically, we chose $\boldsymbol{\mu}_\gamma^* = \mathbf{0}$ and $\boldsymbol{\Sigma}_\gamma^* = \mathbf{I}_k$ as mean and covariance matrix of the multivariate normal prior distribution for $\boldsymbol{\gamma}$, whereas $\nu_h^* = 4$, and $\boldsymbol{V}_h = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ were chosen as the hyperparameter values of the inverse-Wishart distribution on $\boldsymbol{\Sigma}_h, h = 0, 1$. Our sensitivity analyses showed that results were robust to different choices of these parameters. Finally, we applied an EM algorithm MLE using a two-component mixture of bivariate normals to the data (without including covariate information) to determine the hyperparameters $\boldsymbol{\mu}_0^h$ and $\boldsymbol{\Sigma}_{\mu 0}^h$, the mean and covariance matrices of the two Gaussian mixture components. We obtained

$$\boldsymbol{\mu}_0^1 = \begin{pmatrix} \mu_{0g}^1 \\ \mu_{0b}^1 \end{pmatrix} = \begin{pmatrix} 36 \\ 2.57 \end{pmatrix}, \boldsymbol{\mu}_0^2 = \begin{pmatrix} \mu_{0g}^2 \\ \mu_{0b}^2 \end{pmatrix} = \begin{pmatrix} 39 \\ 3.30 \end{pmatrix}, \boldsymbol{\Sigma}_{\mu 0}^1 = \begin{pmatrix} 7.66 & 1.37 \\ 1.37 & 0.35 \end{pmatrix}, \boldsymbol{\Sigma}_{\mu 0}^2 = \begin{pmatrix} 1.34 & 0.19 \\ 0.19 & 0.22 \end{pmatrix}.$$

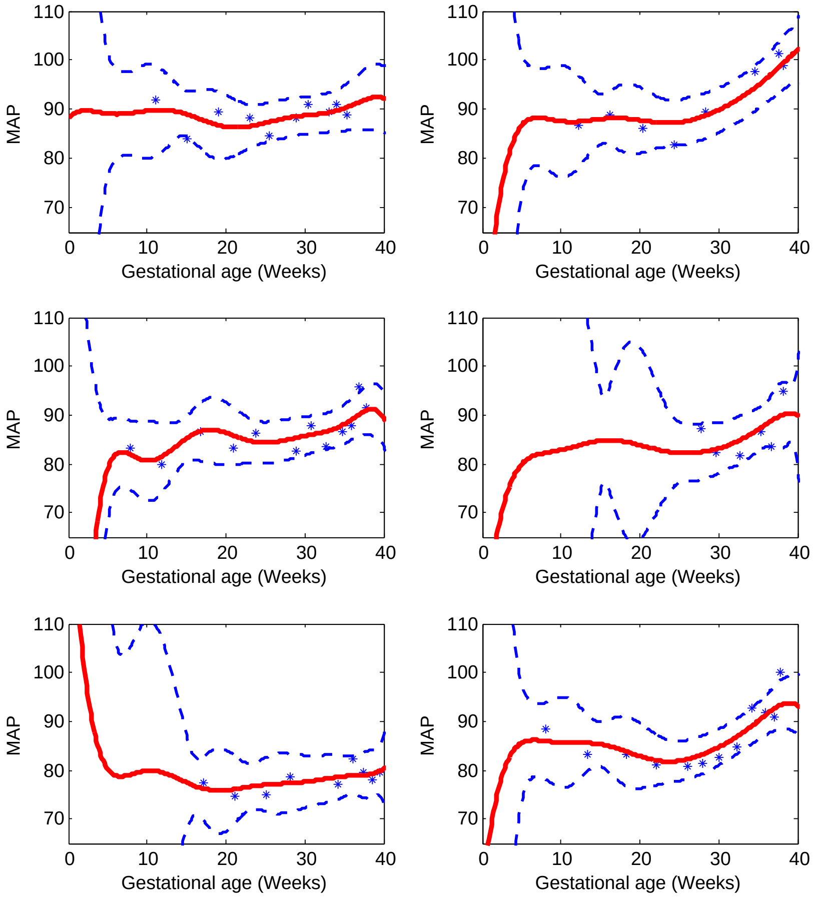


Figure A.1: MAP function estimates obtained with two-stage FPCA approach for 6 randomly selected women in the Healthy Pregnancy, Healthy Baby study. The posterior means are solid lines and dashed lines are 95% pointwise credible intervals. The x-axis scale is time in weeks starting at the estimated day of ovulation.

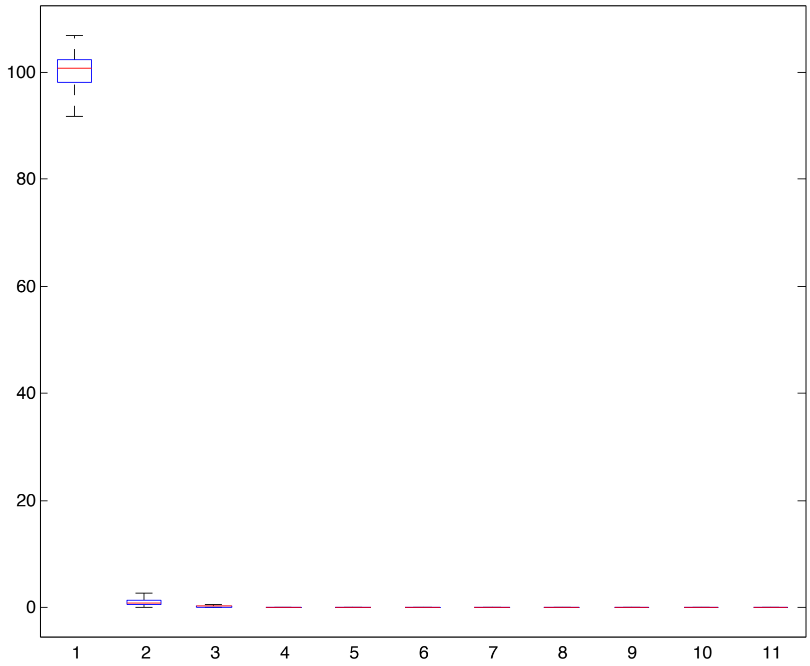


Figure A.2: Side-by-side boxplot of the norms of the posterior mean estimates of the columns of the factor loading matrix $\boldsymbol{\Lambda}$.

The boxplots show a decay of the norms from $\boldsymbol{\lambda}_1$ to $\boldsymbol{\lambda}_{k*}$, as expected by the structure induced by the MGPS prior on $\boldsymbol{\Lambda}$. Note that $k^* = 11$ corresponds to the posterior mean number of factors. Therefore, the first few important factors are loaded heavily and significantly contribute to the estimated of $\boldsymbol{\theta}_i$. Although the norms of the remaining factors appear equal to zero, none of the factor loadings is exactly equal to zero, but shrunk towards zero by the MGPS prior.

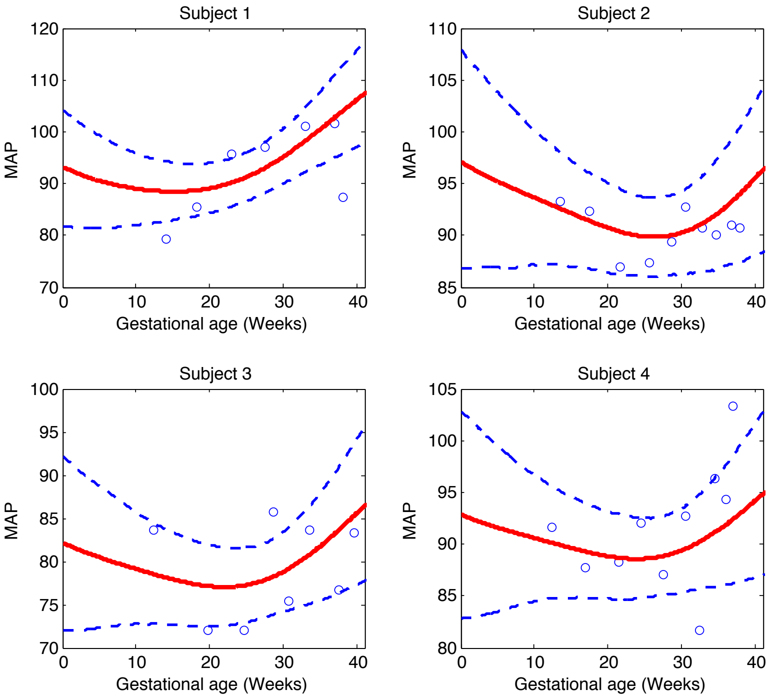


FIGURE A.3: MAP function estimates at the 35*th* week for four subjects in the test set of the Healthy pregnancy Healthy Baby study. The posterior means are solid lines and dashed lines are 95% pointwise credible intervals. The x-axis scale is time in weeks starting at the estimated day of ovulation.

Appendix B

Additional supporting plots for the non-stationary Gaussian process emulator

This appendix presents additional supporting plots that show the performance of the proposed non-stationary Gaussian process emulator as described in Chapter 5 in a variety of examples.

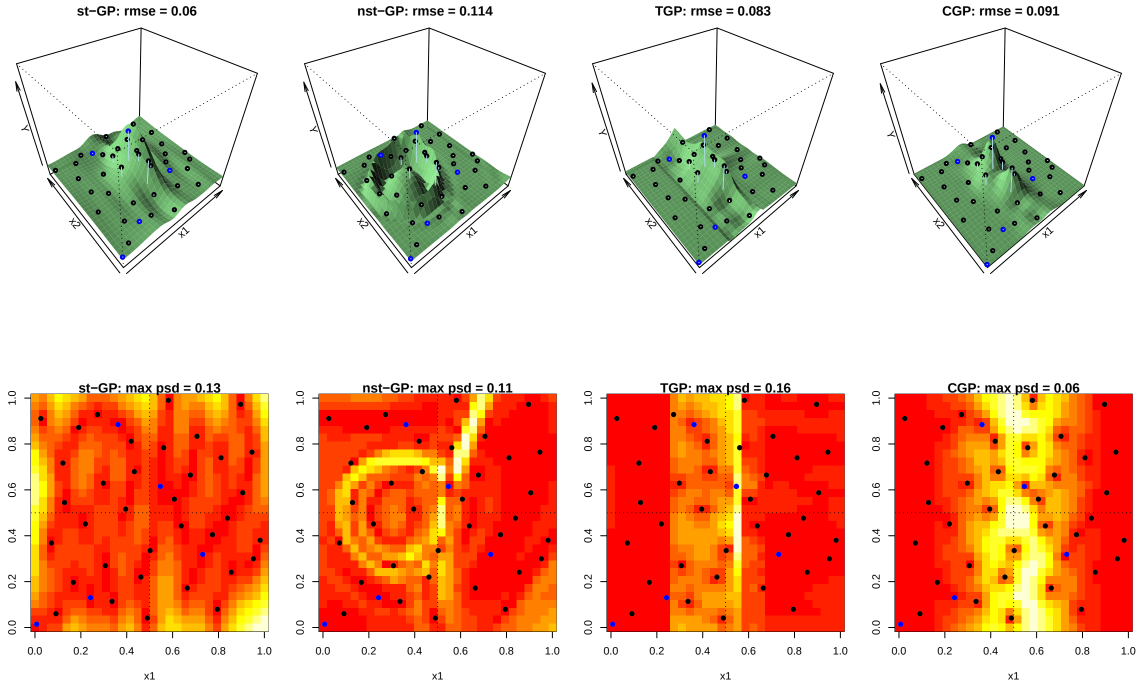


Figure B.1: Menhir function estimates at $T = 40$ (latin hypercube design). Top row: posterior mean predictive surface, $\hat{f}$; bottom row: predictive standard deviation, $\hat{\sigma}$. The quality of the prediction is assessed at a collection of 900 points in $\Omega = [0, 1]^2$, i.e. an expanded grid of 30 equally spaced points along each coordinate axes. Blue points correspond to the initial design used for particle learning for our non-stationary Gaussian process. We also report the root mean squared error (rmse) and the maximum predictive standard deviation (max psd), which are computed based on the test points. Comparison among stationary Gaussian process (st-GP), non-stationary Gaussian process (nst-GP) via latent input augmentation, treed Gaussian process (TGP), and composite Gaussian process (CGP).

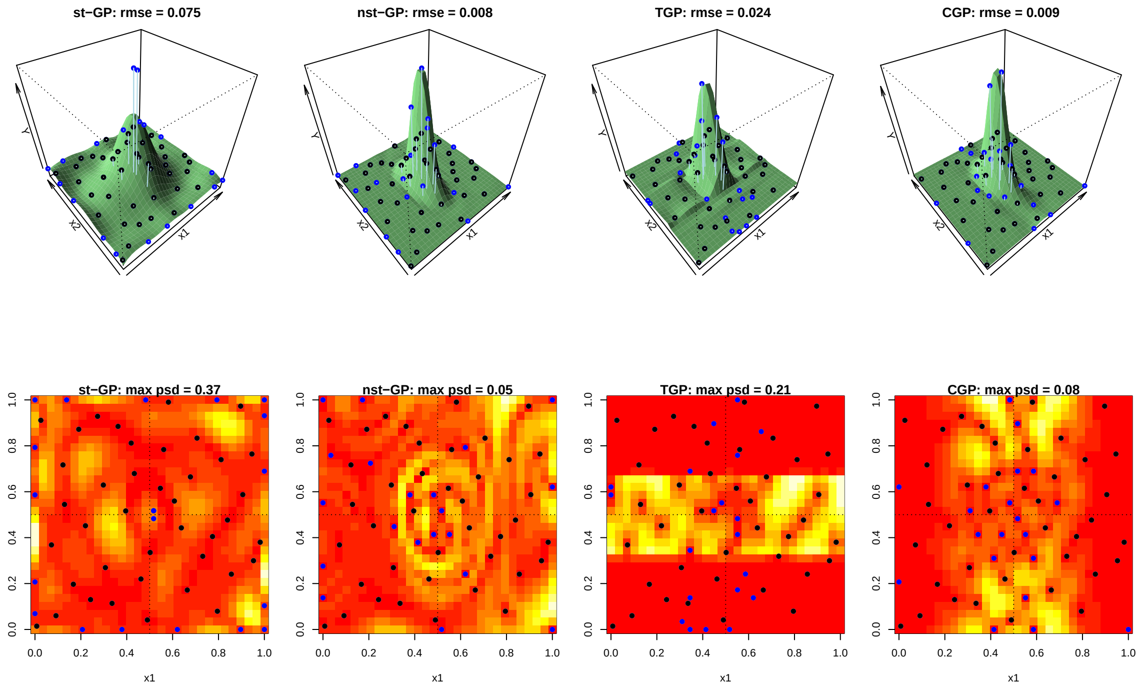


Figure B.2: Menhir function estimates at 60 design points. Black points denote the initial 40 latin hypercube design whereas blue points denote the additional inputs selected via active learning MacKay. Comparison among stationary Gaussian process (st-GP), non-stationary Gaussian process (nst-GP) via latent input augmentation, treed Gaussian process (TGP), and composite Gaussian process (CGP). Note how the stationary Gaussian process does not interpolate at the peak. We defer a discussion on this phenomenon in Section 5.7.

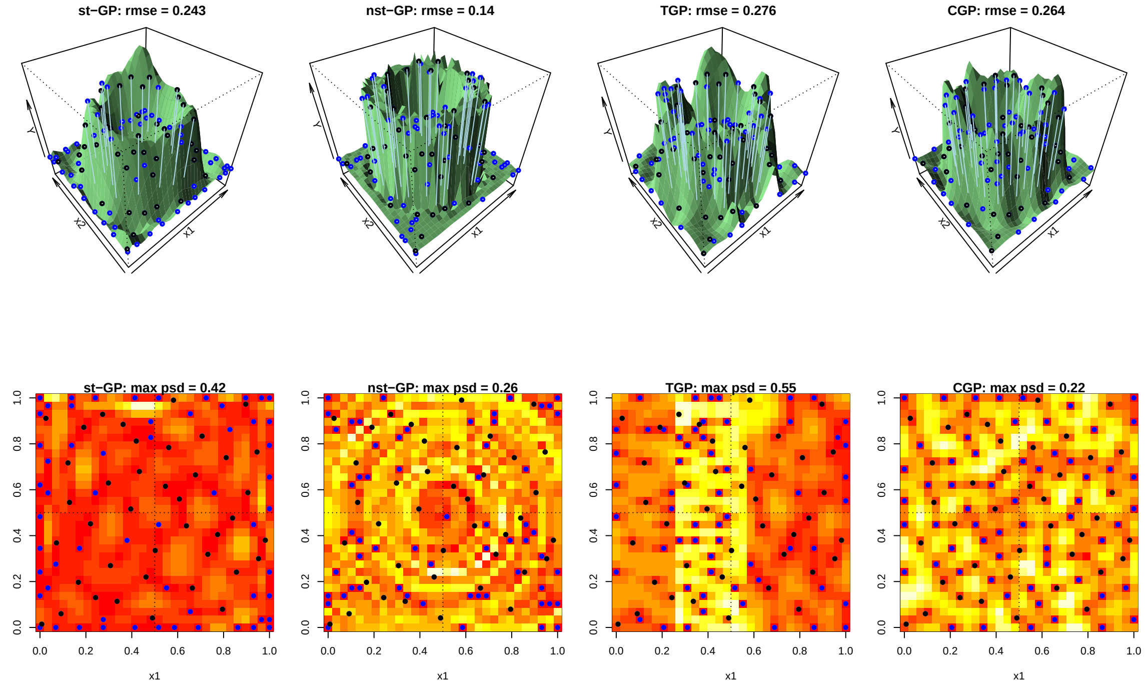


Figure B.3: Well function estimates at 100 design points. Black points denote the initial 40 latin hypercube design whereas blue points denote the additional inputs selected via active learning MacKay. Comparison among stationary Gaussian process (st-GP), non-stationary Gaussian process (nst-GP) via latent input augmentation, treed Gaussian process (TGP), and composite Gaussian process (CGP).

Six-dimensional numerical examples

Recall that $\phi_j \geqslant 0$ ($\tilde{\phi}_j \geqslant 0$) controls the sensitivity of f (g) to x_j. Thus, $\phi_j = 0$ ($\tilde{\phi}_j = 0$) removes x_j (dimension reduction), whereas a larger ϕ_j ($\tilde{\phi}_j$) gives smaller correlation, i.e. $f(\boldsymbol{x})$ and $f(\boldsymbol{x}')$ ($g(\boldsymbol{x})$ and $g(\boldsymbol{x}')$) are less related in the x_j direction and the function is more complex. Figure B.4 shows the distribution across particles of the estimated $\{\tilde{\phi}_i\}_{i=1}^{6}$ in the 6D well example. Note that $\tilde{\phi}_5$ are $\tilde{\phi}_6$ are estimated to be smaller than $\{\tilde{\phi}_i\}_{i=1}^{4}$, thus showing that our emulator is learning that f is less sensitive to these input dimensions. Besides a few large isolated outliers, the correlation length parameters of K are estimated to be small except for ϕ_7, which is associated to the latent input Z (Figure B.5).

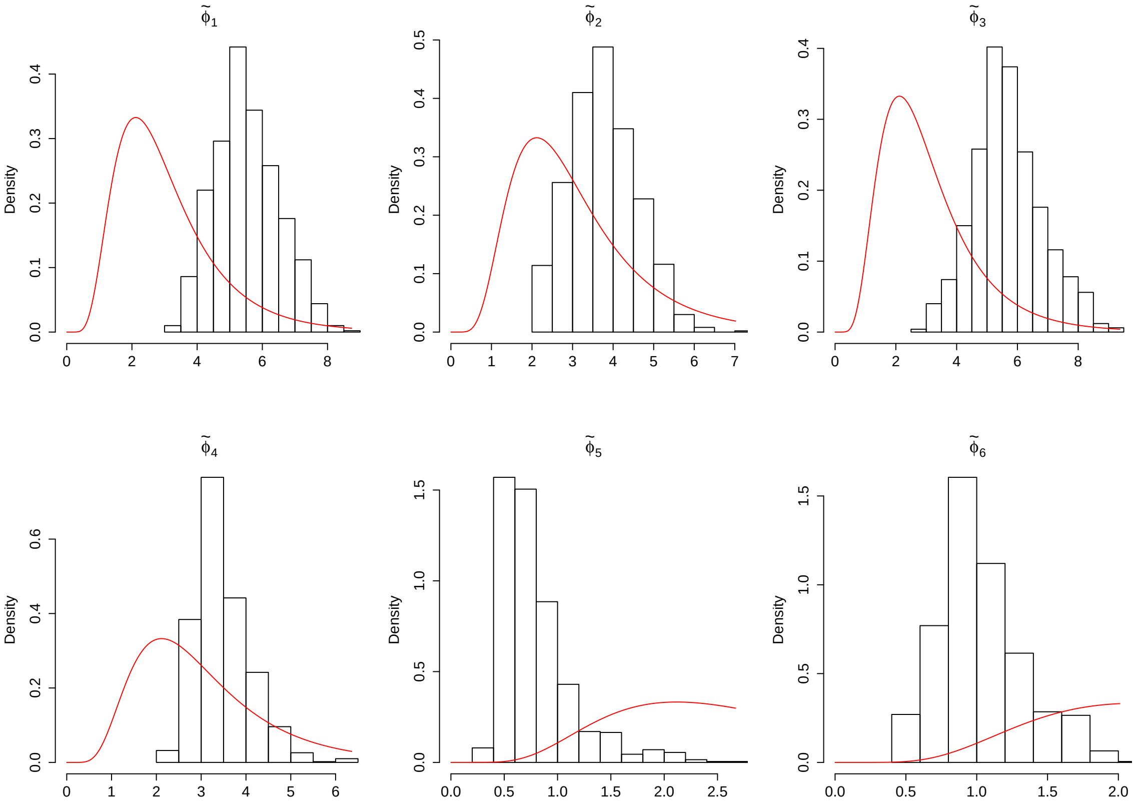


FIGURE B.4: Distribution across particles of the correlation length parameters of $\tilde{K}$ at $T = 200$ in the 6-dimensional well example. The red curve denotes the prior distribution on the latent correlation length parameters $\{\tilde{\phi}_i\}_{i=1}^{6}$. Specifically, $\log \tilde{\phi}_i \sim \mathrm{N}(0.5, 0.25), i = 1, \ldots, 6$.

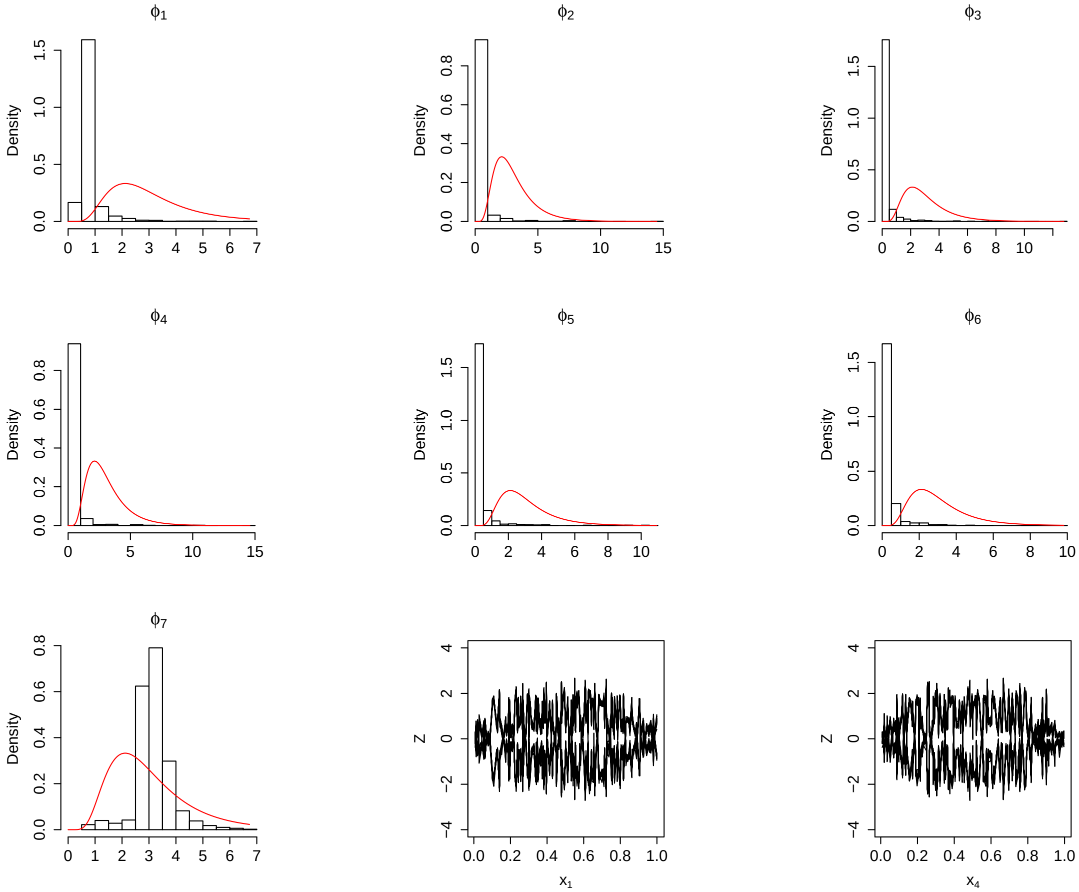


FIGURE B.5: Distribution across particles of the correlation length parameters of K at $T = 200$ in the 6-dimensional well example. The red curve denotes the prior distribution on the correlation length parameters $\{\phi_i\}_{i=1}^{7}$. Specifically, $\log \phi_i \sim \mathrm{N}(1, 0.25), i = 1, \ldots, 7$. At every input configuration $\{(x_1, x_2, \ldots, x_6)_t\}_{t=1}^{T}$ corresponds an estimate of the latent input Z, where $\{x_i\}_{i=1}^{6}$ are known inputs. The last two panels show the estimated latent input Z at each design point (initial 40 latin hypercube design + points selected via active learning MacKay). Z is plotted versus the first and fourth dimension of the corresponding input configuration.

Langley glide-back booster (LGBB) experiment

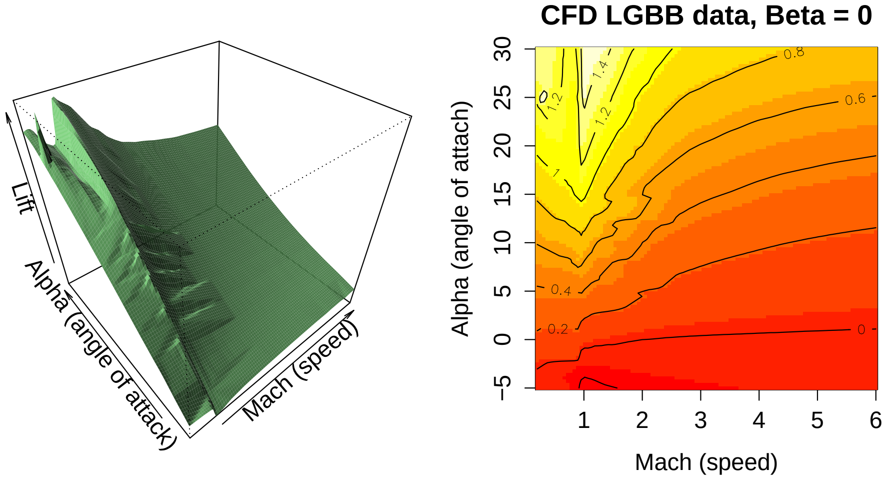


Figure B.6: Interpolated lift surface plotted as a function of Mach (speed) and Alpha (angle of attack) with Beta (side-slip angle) fixed to zero. The ridge at Mach 1 denotes a distinction between subsonic flows and supersonic flows. The upper-left corner of the plot (high angle of attack, low speed) shows a spike which is a result of false convergence of the simulator (Gramacy and Lee, 2008).

Two stage approximation

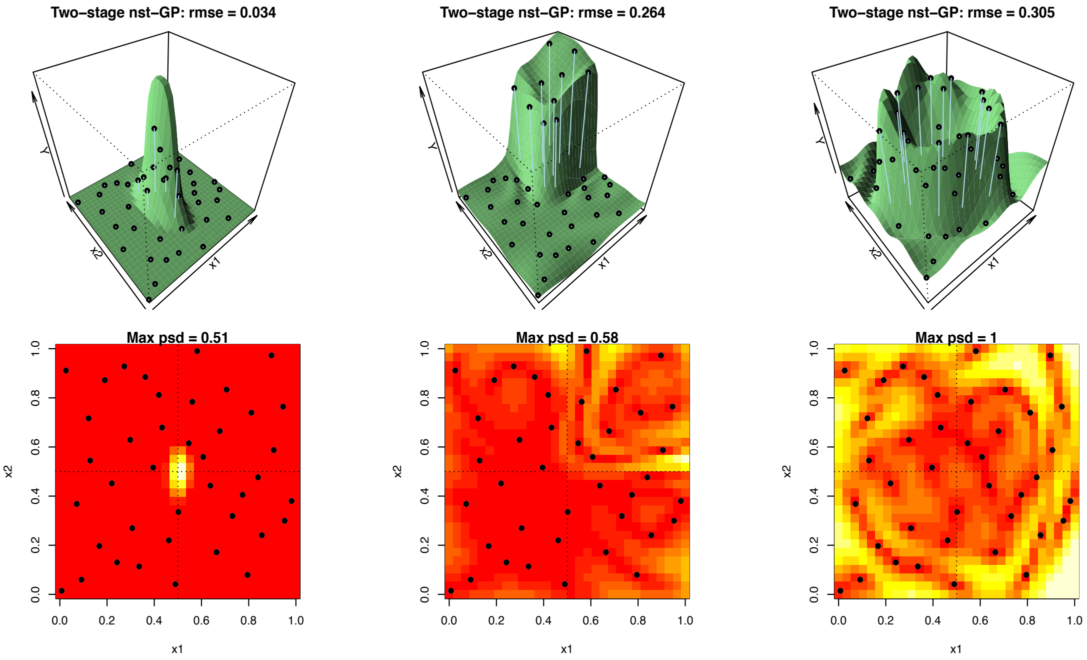


Figure B.7: Predictive surface and standard deviation at a set of 900 predictive points obtained with a two stage Markov chain Monte Carlo-based implementation of the non-stationary Gaussian process emulator on the 2D numerical examples. Quantitative summaries report the root mean squared error (RMSE) and the maximum predictive standard deviation (pred sd) computed based on the test points. The fit is based on the same 40 latin hypercube design (black points) that was used in our 2D numerical examples in Section 5.4.1.

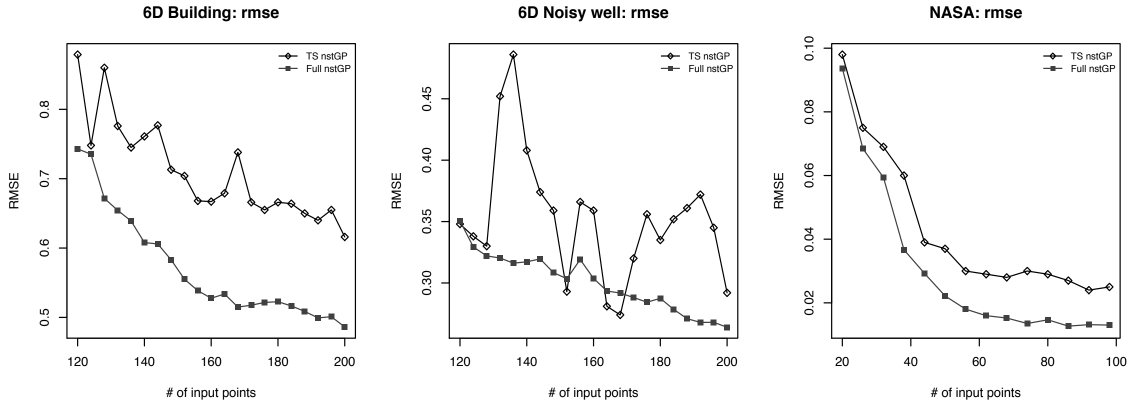


FIGURE B.8: Progression of the root mean square error (RMSE) as additional input points are being selected for the 6D and NASA examples. Comparison between full Bayes non-stationary Gaussian process (Full nstGP) and two stage approximation (TS nstGP).

Bibliography

Ahdesmäki, M., Lähdesmäki, H., Pearson, R., Huttunen, H., and Yli-Harja, O. (2005), "Robust detection of periodic time series measured from biological systems," *BMC Bioinformatics*, 6.

Albert, J. H. and Chib, S. (1993), "Bayesian analysis of binary and polychotomous response data," *Journal of the American Statistical Association*, 88, 669–679.

Anderson, P. E., Smith, J. Q., Edwards, K. D., and Millar, A. J. (2006), "Guided conjugate Bayesian clustering for uncovering rhythmically expressed genes," Working paper.

Andrianakis, I. and Challenor, P. G. (2012), "The effect of the nugget on Gaussian process emulators of computer models," *Computational Statistics and Data Analysis*, 56, 4215–4228.

Arminger, G. (1998), "A Bayesian approach to nonlinear latent variable models using the Gibbs sampler and the Metropolis-Hastings algorithm," *Psychometrika*, 63, 271–300.

Ba, S. and Joseph, R. (2012), "Composite Gaussian process models for emulating expensive functions," *Annals of Applied Statistics*, 6, 1838–1860.

Bartholomew, D. J. and Knott, M. (1999), *Latent variable models and factor analysis*, London: Arnold.

Bastos, L. S. and O'Hagan, A. (2009), "Diagnostics for Gaussian process emulators," *Technometrics*, 51, 425–438.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. (2007), "A framework for validation of computer models," *Technometrics*, 49, 138–154.

Behseta, S., Kass, R. E., and Wallstrom, G. L. (2005), "Hierarchical models for assessing variability among functions," *Biometrika*, 92, 419–434.

Bhattacharya, A. and Dunson, D. B. (2011a), "Sparse Bayesian infinite factor models," *Biometrika*, 98, 291–306.

Bhattacharya, A. and Dunson, D. B. (2011b), "Sparse Bayesian infinite factor models," *Biometrika*, 98, 291–306.

Bigelow, J. L. and Dunson, D. B. (2009), "Bayesian semiparametric joint models for functional predictors," *Journal of the American Statistical Association*, 104, 26–36.

Busby, D. (2009), "Hierarchical adaptive experimental design for Gaussian process emulators," *Reliability Engineering & System Safety*, 94, 1183–1193.

Chudova, D., Ihler, A., Lin, K. K., Andersen, B., and Smyth, P. (2009), "Bayesian detection of non-sinusoidal periodic patterns in circadian expression data," *Bioinformatics*, 25, 3114–3120.

Cohn, D. A. (1996), "Neural network exploration using optimal experiment design," *Neural Networks*, 9, 1071–1083.

Crainiceanu, C. and Goldsmith, J. (2010), "Bayesian functional data analysis using WinBUGS," *Journal of Statistical Software*, 32, 1–33.

Cunningham, F. G., Gant, N. F., Leveno, K. J., Gilstrap, L. C., Hauth, J. C., and Wenstrom, K. D. (2010), *Hypertensive disorders in pregnancy*, In: Williams Obstetrics, McGraw-Hill, New York, 21st edition.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991), "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *Journal of the American Statistical Association*, 86, 953–963.

De la Cruz-Mesia, R., Quintana, F. A., and Müller, P. (2007), "Semiparametric Bayesian classification with longitudinal markers," *Applied Statistics*, 56, 119–137.

Dodd, A. N., Gardner, M. J., Hotta, C. T., Hubbard, K. E., Dalchau, N., Love, J., Assie, J.-M., Robertson, F. C., Jakobsen, M. K., Goncalves, J., Sanders, D., and Webb, A. A. R. (2007), "The *Arabidopsis* circadian clock incorporates a cADPR-based feedback loop," *Science*, 318, 1789–1792.

Dunson, D. B. (2009), "Nonparametric Bayes local partition models for random effects," *Biometrika*, 96, 249–262.

Dunson, D. B. (2010), "Multivariate kernel partition process mixtures," *Statistica Sinica*, 20, 1395–1422.

Duvenaud, D., Nickisch, H., and Rasmussen, C. E. (2011), "Additive Gaussian processes," in *Proceedings of the Conference on Neural Information Processing Systems (NIPS).*

Edwards, K. D., Anderson, P. E., Hall, A., Salathia, N. S., Locke, J. C. W., Lynn, J. R., Straume, M., Smith, J. Q., and Millar, A. J. (2006), "FLOWERING LOCUS C mediates natural variation in the high-temperature response of the *Arabidopsis* circadian clock," *The Plant Cell*, 18, 639–650.

Fan, Y., Ginis, I., Hara, T., Wright, C. W., and Walsh, E. J. (2009), "Numerical simulations and observations of surface wave fields under an extreme tropical cyclone," *Journal of Physical Oceanography*, 39, 2097–2116.

Gamerman, D. (1997), "Sampling from the posterior distribution in generalized linear mixed models," *Statistics and Computing*, 7, 57–68.

Gilks, W. R. and Berzuini, C. (2001), "Following a moving target: Monte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society: Series B*, 63, 127–146.

Gramacy, R. B. and Lee, H. K. H. (2008), "Bayesian treed Gaussian process models with an application to computer modeling," *Journal of the American Statistical Association*, 103, 1119–1130.

Gramacy, R. B. and Lee, H. K. H. (2009), "Adaptive design and analysis of supercomputer experiments," *Technometrics*, 51, 130–145.

Gramacy, R. B. and Polson, N. G. (2011), "Particle learning of Gaussian process models for sequential design and optimization," *Journal of Computational and Graphical Statistics*, 20, 102–118.

Hamilton, B. E., Martin, J. A., and Ventura, S. J. (2010), "Births: preliminary data for 2008," *National Vital Statistic Reports*, 58, 1–17.

Heard, N. A., Holmes, C. C., and Stephens, D. A. (2006), "A quantitative study of gene regulation involved in the immune response of anopheline mosquitoes: an application of Bayesian hierarchical clustering of curves," *Journal of the American Statistical Association*, 101, 18–29.

Hughes, M. E., Hogenesch, J. B., and Kornacker, K. (2010), "JTK_CYCLE: an efficient nonparametric algorithm for detecting rhythmic components in genome-scale data sets," *Journal of Biological Rhythms*, 25, 372–380.

James, G. and Sugar, C. (2003), "Clustering for sparsely sampled functional data," *Journal of the American Statistical Association*, 98, 397–408.

James, G. M., Hastie, T. J., and Sugar, C. A. (2000), "Principal components models for sparse functional data," *Biometrika*, 87, 587–602.

Jiang, C. R. and Wang, J. L. (2010), "Covariate adjusted functional principal components analysis for longitudinal data," *Annals of Statistics*, 38, 1194–1226.

Jones, B. L., Nagin, D. S., and Roeder, K. (2001), "A SAS procedure based on mixture models for estimating developmental trajectories," *Sociological Methods and Research*, 29, 374–393.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998), "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, 13, 455–492.

Jouffe, C., Cretenet, G., Symu, L., Martin, E., Atger, F., Naef, F., and Gachon, F. (2013), "The circadian clock coordinates ribosome biogenesis," *PLoS Science*, 11.

Kennedy, M. C. and O'Hagan, A. (2001), "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B*, 63, 425–464.

Kim, H.-M., Mallick, B. K., and Holmes, C. (2005), "Analyzing nonstationary spatial data using piecewise Gaussian processes," *Journal of the American Statistical Association*, 100, 653–668.

Lopes, H. F. and West, M. (2004), "Bayesian model assessment in factor analysis," *Statistica Sinica*, 14, 41–67.

Lopes, H. F., Carvalho, C. M., Johannes, M. S., and Polson, N. G. (2011), *Particle learning for sequential Bayesian computation*, Oxford University Press.

Luan, Y. and Li, H. (2003), "Clustering of time-course gene expression data using a mixed-effects model with B-splines," *Bioinformatics*, 19, 474–482.

MacKay, D. J. (1992), "Information-Based Objective Functions for Active Data Selection," *Neural Computation*, 4, 590–604.

Matsuo, T., Yamaguchi, S., Mitsui, S., Emi, A., Shimoda, F., and Okamura, H. (2003), "Control mechanism of the circadian clock for timing of cell division in vivo," *Science*, 302, 255–259.

Moustaki, I. and Knott, M. (2000), "Generalized latent trait models," *Psychometrika*, 65, 391–411.

Müller, P., Parmigiani, G., Robert, C., and Rousseau, J. (2004), "Optimal sample size for multiple testing: the case of gene expression microarrays," *Journal of the American Statistical Association*, 99, 990–1001.

Müller, P., Parmigiani, G., and Rice, K. (2006), "FDR and Bayesian multiple comparisons rules," *Johns Hopkins University, Dept. of Biostatistics Working Papers.*

Murray, I., Adams, R. P., and MacKay, D. J. C. (2010), "Elliptical slice sampling," *Journal of Machine Learning Research*, 9, 541–548.

Nagin, D. S. (1999), "Analyzing developmental trajectories: a semiparametric group-based approach," *Psychological Methods*, 4, 139–157.

Nakajima, J. and West, M. (2013), "Bayesian analysis of latent threshold dynamic models," *Journal of Business and Economic Statistics*, 31, 151–164.

Nelson, W., Tong, Y. L., Lee, J. K., and F., H. (1979), "Methods for cosinor-rhythmometry," *Chronobiologia*, 6, 305–323.

Newton, M. A., Noueiry, A., Sarkar, D., and Ahlquist, P. (2004), "Detecting differential gene expression with a semiparametric hierarchical mixture method," *Biostatistics*, 5, 155–176.

Noh, M. and Lee, Y. (2007), "Robust modeling for inference from generalized linear model classes," *Journal of the American Statistical Association*, 102, 1059–1072.

O'Brien, S. M. and Dunson, D. B. (2004), "Bayesian multivariate logistic regression," *Biometrics*, 60, 739–746.

Paciorek, C. J. and Schervish, M. J. (2004), "Nonstationary covariance functions for Gaussian process regression," in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, vol. 16, pp. 273–280, MIT Press.

Paciorek, C. J. and Schervish, M. J. (2006), "Spatial modelling using a new class of nonstationary covariance functions," *Environmetrics*, 17, 483–506.

Petrone, S., Guindani, M., and Gelfand, A. E. (2009), "Hybrid Dirichlet mixture models for functional data," *Journal of the Royal Statistical Society: Series B*, 71, 755–782.

Pfingsten, T., Kuss, M., and Rasmussen, C. E. (2006), "Nonstationary Gaussian process regression using a latent extension of the input space," in *Presented as a poster at the ISBA Eight World Meeting on Bayesian Statistics*.

Phillips, M. L. (2009), "Circadian rhythms: Of owls, larks and alarm clocks," *Nature*, 458, 142–144.

Ramsey, J. O. and Silverman, B. W. (2005), *Functional Data Analysis*, 2nd edition. New York: Springer - Verlag.

Rasmussen, C. E. and Ghahramani, Z. (2001), "Infinite mixtures of Gaussian process experts," in *Advances in Neural Information Processing Systems*, vol. 14, pp. 881–888, MIT Press.

Ray, S. and Mallick, B. (2006), "Functional clustering by Bayesian wavelet methods," *Journal of the Royal Statistical Society: Series B*, 68, 305–322.

Reiss, P. T., Huang, L., and Mennes, M. (2010), "Fast function-on-scalar regression with penalized basis expansions," *The International Journal of Biostatistics*, 6, 1–30.

Rice, J. A. (2004), "Functional and longitudinal data analysis: perspectives on smoothing," *Statistica Sinica*, 14, 631–647.

Rice, J. A. and Silverman, B. W. (1991), "Estimating the mean and covariance structure nonparametrically when the data are curves," *Journal of the Royal Statistical Society: Series B*, 53, 233–243.

Ridgeway, G. and Madigan, D. (2003), "A sequential Monte Carlo method for Bayesian analysis of massive datasets," *Journal of Knowledge Discovery and Data Mining*, 7, 301–319.

Roberts, G. O. and Rosenthal, J. S. (2007), "Coupling and ergodicity of adaptive MCMC," *Journal of Applied Probability*, 44, 458–475.

Rodriguez, A., Dunson, D. B., and Gelfand, A. E. (2009), "Bayesian nonparametric functional data analysis through density estimation," *Biometrika*, 98, 149–210.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and analysis of computer experiments," *Statistical Science*, 4, 409–423.

Sammel, M. D., Ryan, L. M., and Legler, J. M. (1997), "Latent variable models for mixed discrete and continuous outcomes," *Journal of the Royal Statistical Society, Ser. B*, 59, 667–678.

Sampson, P. D. and Guttorp, P. (1992), "Nonparametric estimation of nonstationary spatial covariance structure," *Journal of the American Statistical Association*, 87, 108–119.

Santner, T., Williams, B., and Notz, W. (2003), *The Design and Analysis of Computer Experiments*, Springer Series in Statistics, Springer.

Schade, L. and Emanuel, K. (1999), "The ocean's effect on the intensity of tropical cyclones: results from a simple coupled atmosphere-ocean model," *Journal of the Atmospheric Science*, 56, 642–651.

Schmidt, A. M. and O'Hagan, A. (2000), "Bayesian inference for nonstationary spatial covariance structure via spatial deformations," *Journal of the Royal Statistical Society: Series B*, 65, 745–758.

Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000), "Gaussian process regression: active data selection and test point rejection," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, vol. 3, pp. 241–246, IEEE.

Straif, K., Baan, R., Grosse, Y., Secretan, B., El Ghissassi, F., Bouvard, V., Altieri, A., Benbrahim-Tallaa, L., and Cogliano, V. (2007), "Carcinogenicity of shift-work, painting, and fire-fighting," *The Lancet Oncology*, 8, 1065–1066.

Straume, M. (2004), "DNA microarray time series analysis: automated statistical assessment of circadian rhythms in gene expression patterning," *Methods in enzymology*, 383, 149–166.

Textor, C., Graf, H., Longo, A., Neri, A., Ongaro, T. E., Papale, P., Timmreck, C., and Ernst, G. G. J. (2009), "Numerical simulation of explosive volcanic eruptions from the conduit flow to global atmospheric scales," *Annals of Geophysics*, 48, 817–842.

van der Vaart, A. and van Zanten, J. (2009), "Adaptive Bayesian estimation using a Gaussian random field with inverse gamma bandwidth," *The Annals of Statistics*, 37, 2655–2675.

Wakefield, J., Zhou, C., and Self, S. (2003), "Modeling gene expression over time: curve clustering with informative prior distributions," in *Bayesian Statistics 7*, eds. J. M. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. F. M. Smith, and M. West, Oxford University Press.

Wedel, M. and Kamakura, W. A. (2001), "Factor analysis with (mixed) observed and latent variables in the exponential family," *Psychometrika*, 66.

West, M. (1984), "Outlier models and prior distributions in Bayesian linear regression," *Journal of the Royal Statistical Society, Ser. B*, 46, 431–439.

Wichert, S., Fokianos, K., and Strimmer, K. (2004), "Identifying periodically expressed transcripts in microarray time series data," *Bioinformatics*, 20, 5–20.

Yao, F., Müller, H. G., and Wang, J. L. (2005), "Functional data analysis for sparse longitudinal data," *Journal of the American Statistical Association*, 100, 577–590.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., and Ruzzo, W. L. (2001), "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, 17, 977–987.

Zhu, H., Vannucci, M., and Cox, D. D. (2011), "A Bayesian hierarchical model for classification with selection of functional predictors," *Biometrics*, 66, 463–473.

Biography

Silvia Montagna was born in Broni (Pavia), Italy on July 20, 1985. She completed her bachelor degree with honors in Economics from Universita' degli Studi di Pavia, Pavia, Italy in 2007. In July 2009, she obtained her Master of Arts in Economics from Universita' degli Studi di Torino. She then came to Duke University in Durham, NC, United States, pursuing a Ph.D. degree in the department of Statistical Science, advised by Professor Surya T. Tokdar. She earned an MS in Statistical Science en route to her Ph.D. in 2013. After completion of her Ph.D., she will be joining the Department of Statistics at Warwick University, Coventry, UK, as a post-doctoral research fellow.